FinalProject

AUTHOR
Jayden Cruz & Christopher Garcia

Bayesian Statistics Final Project

In this project, we will explore the Fish Market Data set, which provides detailed information about different fish species and their physical characteristics. The data set include the following variables:

- **Species**: The species of the fish (categorical).
- **Weight**: The weight of the fish in grams (continuous).
- **Length1(Vertical), Length2(Diagonal), Length3(Cross-Length)**: Various measurements of length in centimeters (continuous).
- **Height**: The height of the fish as a percentage of its cross-sectional length (continuous).
- **Width**: The width of the fish as a percentage of its cross-sectional length (continuous).

This data set was collected form Kaggle.com (https://www.kaggle.com/datasets/vipullrathod/fish-market/data)

Our goal is to model and analyze the relationships between these variables to predict the **Weight** of the fish, a continuous outcome. This will involve the following steps:

1. Developing **three Directed Acyclic Graph (DAG) models**, where we explore one predictor at a time and a combined predictor approach:
    - A model for Weight with one continuous variable Length 2.
    - A model for Weight with one another continuous variable Heigth.
    - A model for Weight with both continuous and categorical predictors.
2. Using **quadratic approximation (quap)** to create and analyze three quadratic models, informed by priors with scientific or logical backing.
3. Re-analyzing the models with **Markov Chain Monte Carlo (MCMC)** techniques to ensure robustness.
4. Comparing the models through metrics like WAIC (Widely Applicable Information Criterion) and visualizations to identify the best-fitting model.

Through this analysis, we aim to understand the predictive power of different fish characteristics and to evaluate the differences between approximation and MCMC-based modeling.

Load in Fish Market Data

```
library(rethinking)
```

```
Loading required package: cmdstanr

Warning: package 'cmdstanr' was built under R version 4.4.2

This is cmdstanr version 0.8.1

- CmdStanR documentation and vignettes: mc-stan.org/cmdstanr

- CmdStan path: /Users/jaydencruz/.cmdstan/cmdstan-2.35.0

- CmdStan version: 2.35.0

Loading required package: posterior

This is posterior version 1.6.0


Attaching package: 'posterior'

The following objects are masked from 'package:stats':

    mad, sd, var

The following objects are masked from 'package:base':

    %in%, match

Loading required package: parallel

rethinking (Version 2.42)


Attaching package: 'rethinking'

The following object is masked from 'package:stats':

    rstudent
```

```
data <- read.csv("Fish.csv")
d <- data
d <- na.omit(d)

#head(d)
#tail(d)
```

The dataset has been loaded in, can see the head and the tail of the data.

Standardize the Variables

```
        d$W <- standardize(d$Weight)
        d$L <- standardize(d$Length2)
        d$H <- standardize(d$Height)

        dcc <- d[ complete.cases(d$W,d$L,d$H) , ]

        head(dcc)
```

```
  Species Weight Length1 Length2 Length3  Height  Width           W          L
1   Bream    242    23.2    25.4    30.0 11.5200 4.0200 -0.43669241 -0.2814139
2   Bream    290    24.0    26.3    31.2 12.4800 4.3056 -0.30260608 -0.1974299
3   Bream    340    23.9    26.5    31.1 12.3778 4.6961 -0.16293282 -0.1787668
4   Bream    363    26.3    29.0    33.5 12.7300 4.4555 -0.09868311  0.0545221
5   Bream    430    26.5    29.0    34.0 12.4440 5.1340  0.08847906  0.0545221
6   Bream    450    26.8    29.7    34.7 13.6024 4.9274  0.14434837  0.1198430
          H
1 0.5946997
2 0.8186739
3 0.7948300
4 0.8770005
5 0.8102749
6 1.0805371
```

Weight is our Y variable.

Length2 is our X1 variable and Height is our X2 variable. We standardize our variables to ensure all variables are on a equal scale as height is in centimeter and weight is in grams.

Quadratic Approximation

DAG Models

0. Importing required libraries

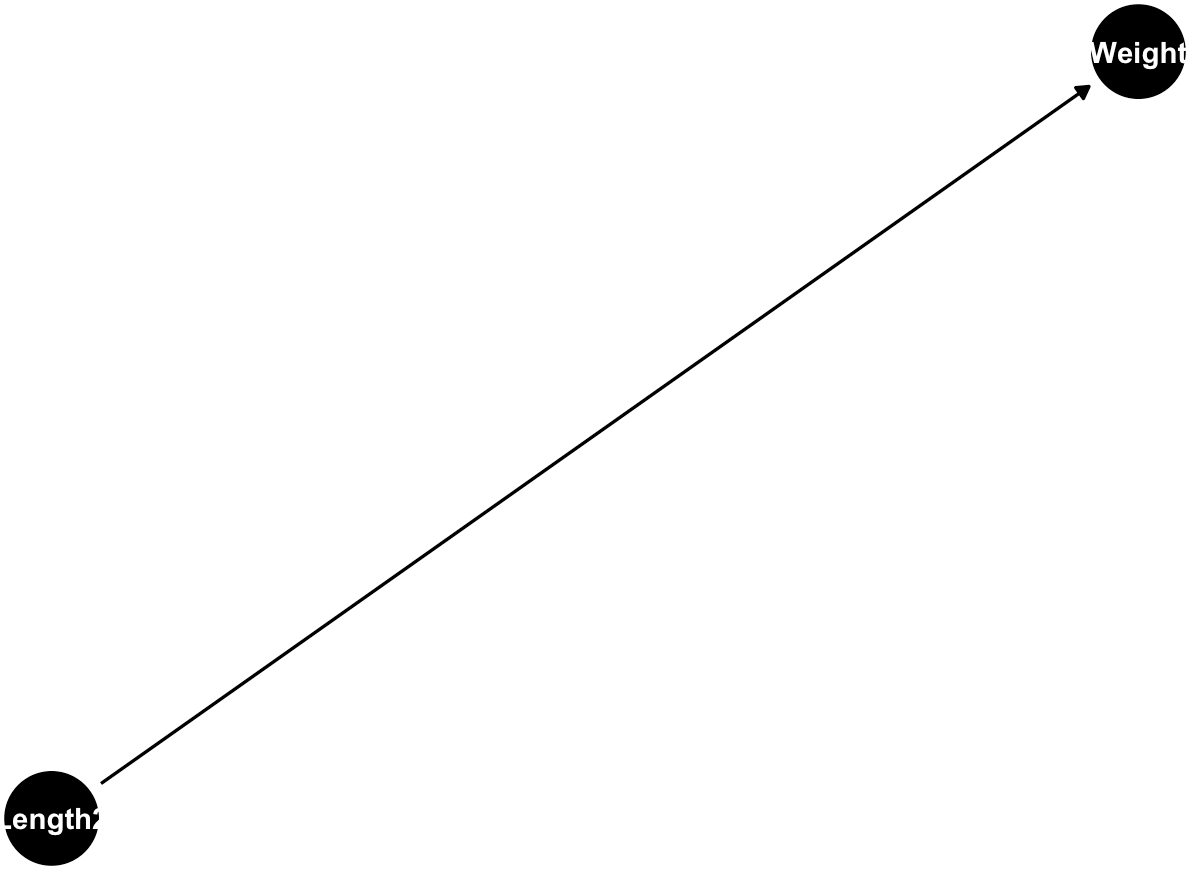
```
        library(ggdag)
```

```
Attaching package: 'ggdag'

The following object is masked from 'package:stats':

    filter
```

```
        library(ggplot2)
        theme_set(theme_dag())
```

1. A model for Y(Weight) with X1(Length 2).

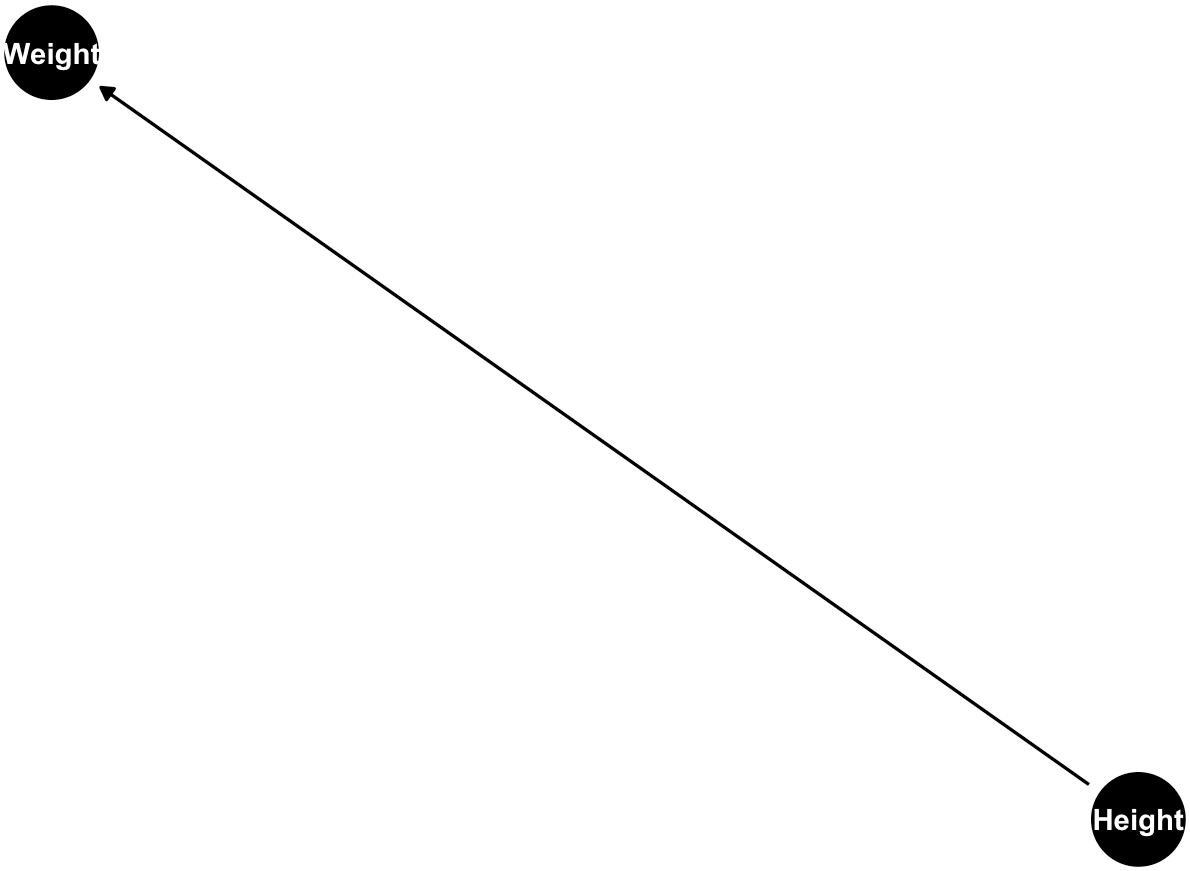
```
        dagify(Weight ~ Length2) %>%
          ggdag()
```



Arrow from length 2 to weight, showing that length 2 can influence the weight of a fish

2. A model for Y(Weight) with X2(Species).

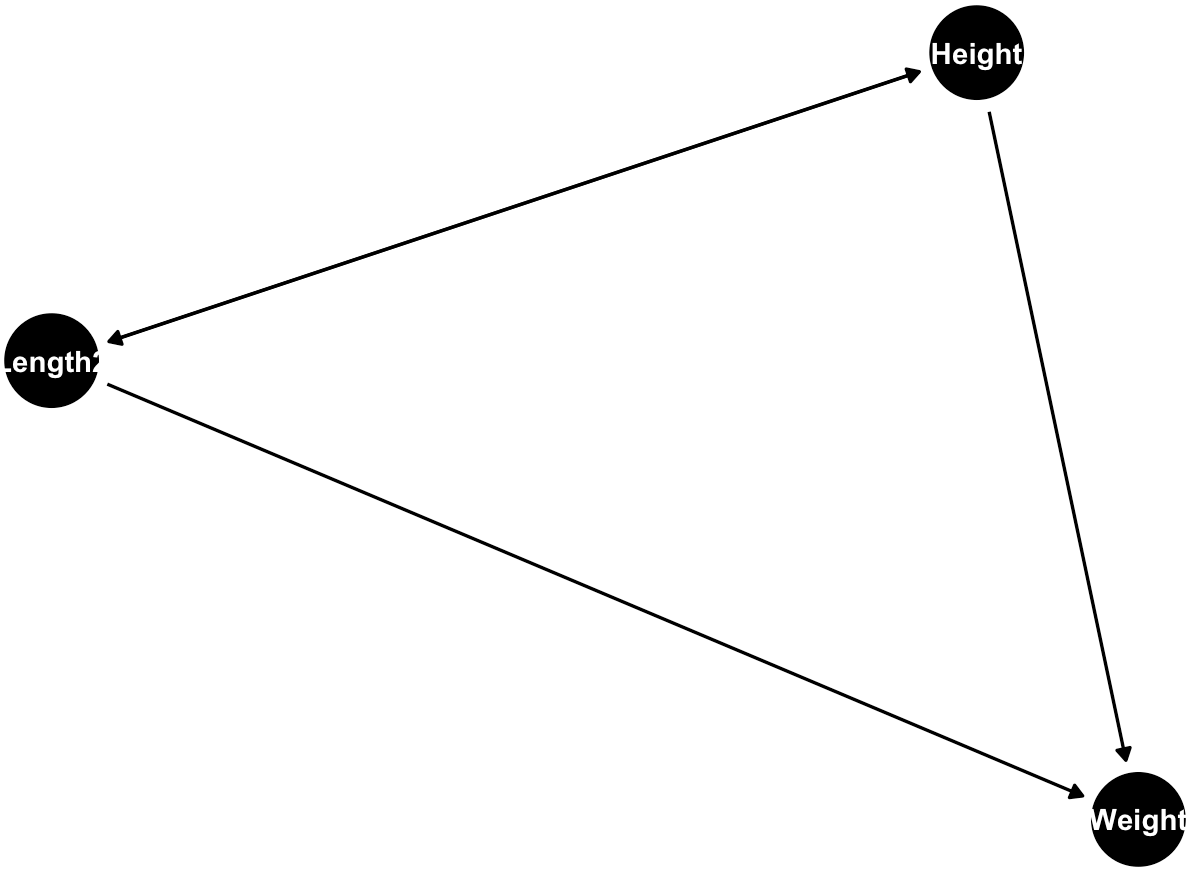
```
        dagify(Weight ~ Height) %>%
          ggdag()
```



Arrow from species to weight, showing that species can influence the weight of a fish

3. A model for Y with X1 and X2.

```
        dagify(
          Weight ~ Height,
          Weight ~ Length2,
          Length2 ~ Height,
          Height ~ Length2
        ) %>%
          ggdag()
```



Arrow from species to weight and length 2 to weight, showing that species and length can influence the weight of a fish

Quadratic Models

Selecting Priors for Weight

```
        average_weight <- mean(dcc$W, na.rm = TRUE)

        print(paste("The average weight of the fish is:", average_weight, "grams"))
```

```
[1] "The average weight of the fish is: -1.67580833905684e-16 grams"
```

```
        weight_sd <- sd(dcc$W, na.rm = TRUE)
        print(paste("STD:", weight_sd))
```

```
[1] "STD: 1"
```

The average for the Weight column after being standardized is -1.675e-16 which is extremely close to 0 so for our prior for weight we will choose 0 with a standard deviation of 1.

Model 1: Length 2 -> Weight

```
m1 <- quap(
 alist(
   W ~ dnorm(mu, sigma),
   mu <- a + bL * L,
   a ~ dnorm(0, 1),
   bL ~ dnorm(0.5, 2),
   sigma ~ dexp(1)
 ), data = d
)
precis(m1)
```

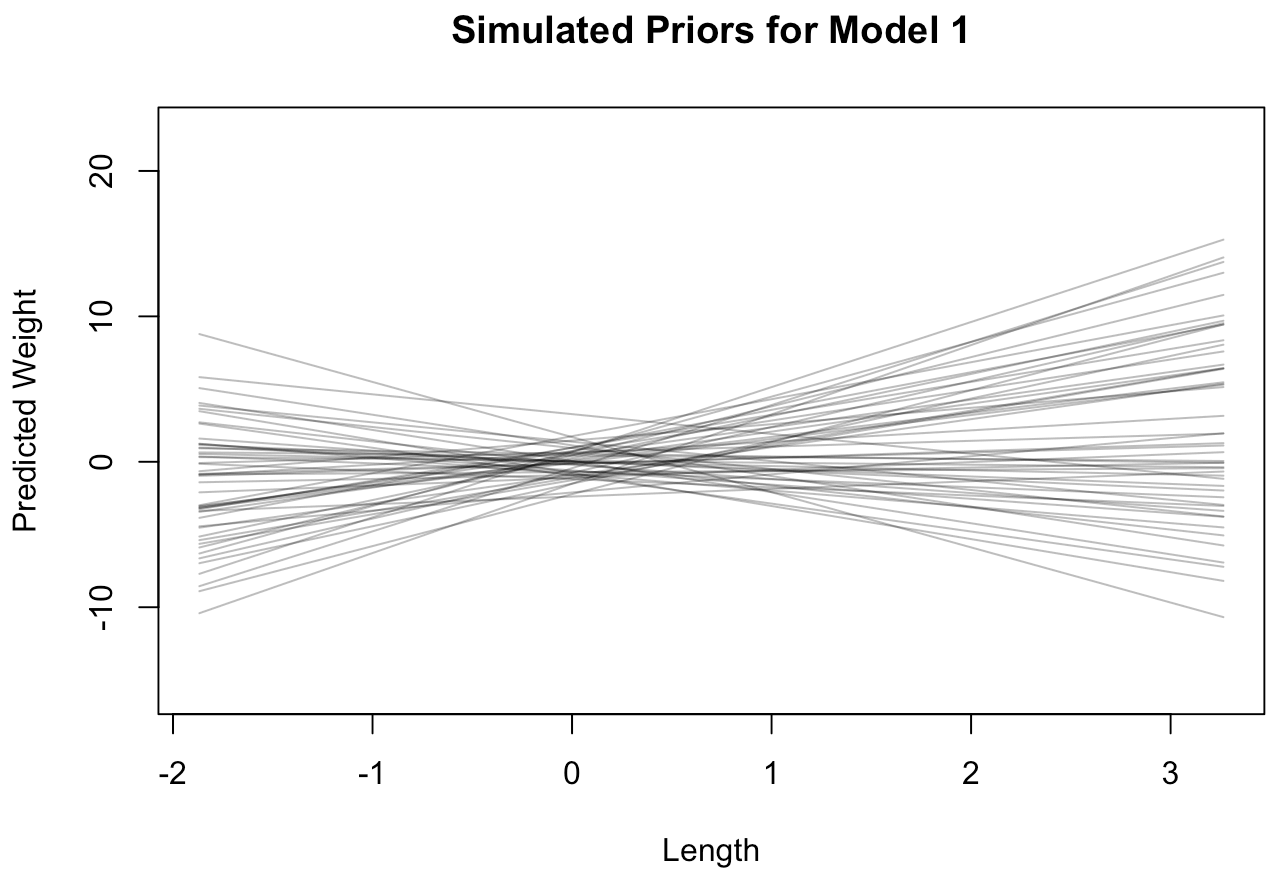
```
               mean         sd        5.5%      94.5%
a     -9.353273e-08 0.03118487 -0.04983954 0.04983936
bL     9.185158e-01 0.03129480  0.86850063 0.96853089
sigma  3.934176e-01 0.02202040  0.35822479 0.42861048
```

Here's a description of **Model 1** based on the new priors:

Model 1 predicts Weight (W) using Length (L). Weight is modeled with a normal distribution, where the mean is a linear function of Length, with an intercept (a) and slope (bL). The prior for the intercept (a) is centered around 0, with a standard deviation of 10. This indicates a belief that the baseline weight is approximately 0, with a fairly wide range of uncertainty. The prior for the slope (bL) is centered at 0.5, with a standard deviation of 2, suggesting that for each 1-unit increase in length, the fish's weight is expected to increase by about 0.5 grams, with some uncertainty regarding the exact effect. The prior for sigma (the weight variability) follows an exponential distribution with a rate of 1, reflecting the natural variability in fish weight. This model captures the relationship between length and weight, accounting for both the expected effect of length and the natural variation in weight.

Model 1: Simulate the priors

```
prior <- extract.prior(m1)

xseq <- seq(min(dcc$L), max(dcc$L), length.out = 100)

mu <- link(m1, post = prior, data = list(L = xseq))

plot(NULL, xlim = range(xseq), ylim = range(mu),
     xlab = "Length", ylab = "Predicted Weight", main = "Simulated Priors for Moc

for ( i in 1:50 ) lines( xseq , mu[i,] , col=col.alpha("black",0.3) )
```



Model 2: Height -> Weight

```
m2 <- quap(
  alist(
    W ~ dnorm(mu, sigma),
    mu <- a + bH * H,
    a ~ dnorm(0, 1),
    bH ~ dnorm(0.5, 2),
    sigma ~ dexp(1)
  ), data = d
)
precis(m2)
```

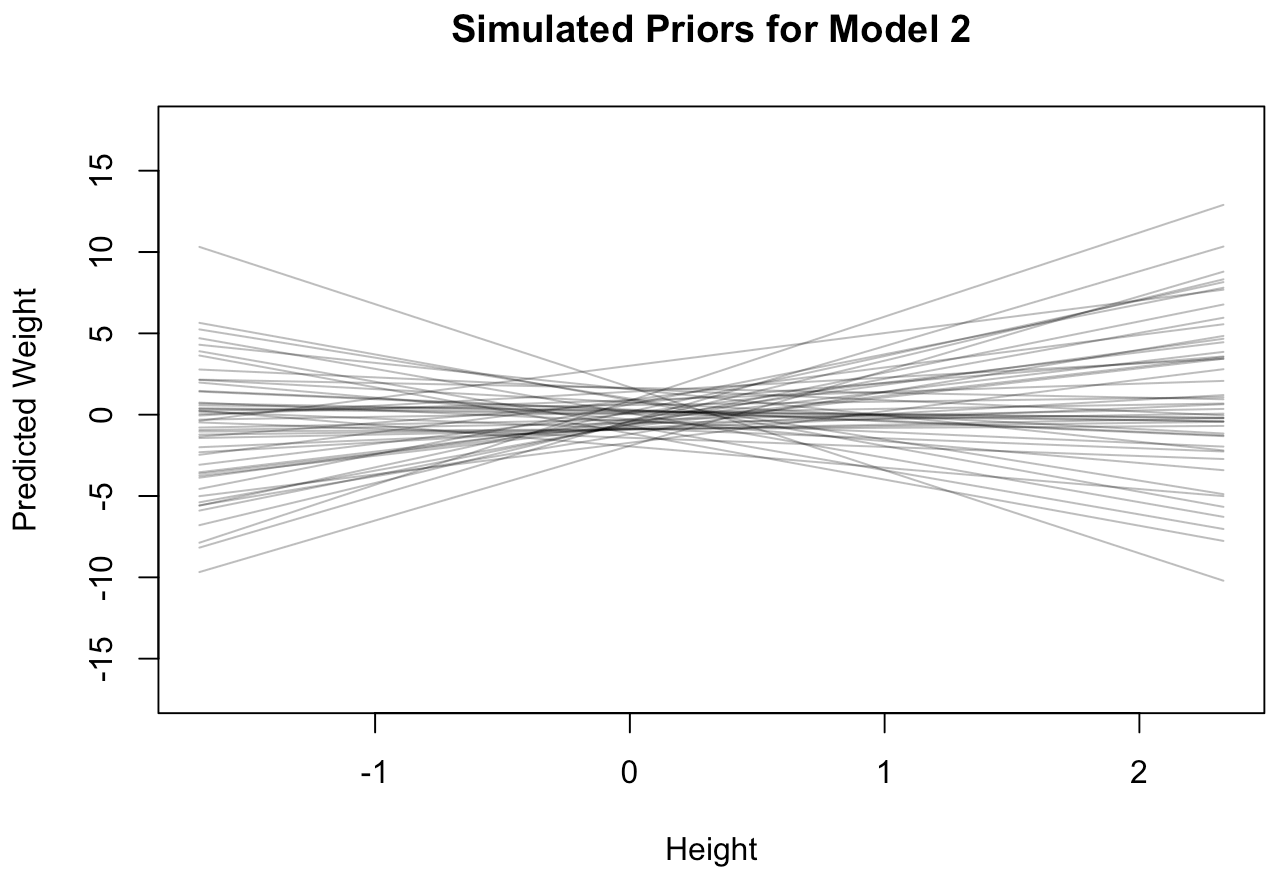
```
              mean         sd       5.5%      94.5%
a     -5.133453e-06 0.05430643 -0.0867973 0.08678703
bH     7.241861e-01 0.05453824  0.6370235 0.81134875
sigma  6.857900e-01 0.03833315  0.6245263 0.74705382
```

Model 2 predicts Weight (W) using Height (H). Weight is modeled with a normal distribution, where the mean is a linear function of Height, with an intercept (a) and slope (bH). The prior for the intercept (a) is centered around 0, with a standard deviation of 10, suggesting that the baseline weight is approximately 0, with a relatively wide range of uncertainty. The prior for the slope (bH) is centered around 0.5, with a standard deviation of 2, implying that for each 1-unit increase in height, the fish's weights is expected to

increase by about 0.5 grams, with some uncertainty. The prior for sigma (the weight variability) follows an exponential distribution with a rate of 1, reflecting the natural variability in fish weight. This model captures the relationship between height and weight, accounting for both the expected effect of length and the natural variation in weight.

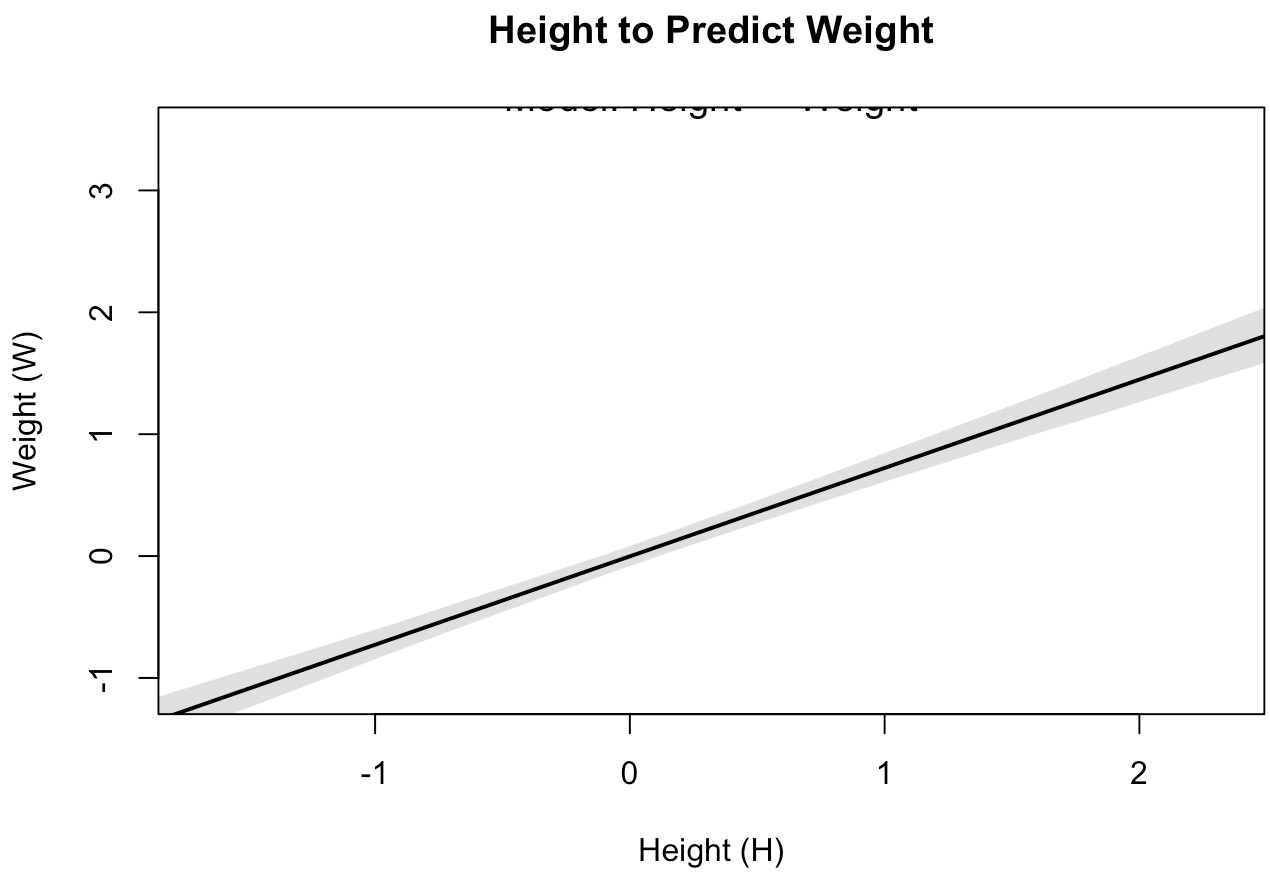
Model 2: Simulate the priors

```
prior <- extract.prior(m2)

xseq <- seq(min(dcc$H), max(dcc$H), length.out = 100)

mu <- link(m2, post = prior, data = list(H = xseq))

plot(NULL, xlim = range(xseq), ylim = range(mu),
     xlab = "Height", ylab = "Predicted Weight", main = "Simulated Priors for Moc

for (i in 1:50) {
  lines(xseq, mu[i,], col = col.alpha("black", 0.3))
}
```



Model 2: Counterfactual Plot

```
xseq <- seq(from = min(dcc$H) - 2, to = max(dcc$H) + 2, length.out = 30)

mu <- link(m2, data = data.frame(H = xseq))

mu_mean <- apply(mu, 2, mean)
mu_PI <- apply(mu, 2, PI)

plot(NULL, xlim = range(dcc$H), ylim = range(dcc$W),
     xlab = "Height (H)", ylab = "Weight (W)", main = "Height to Predict Weight")

lines(xseq, mu_mean, lwd = 2)

shade(mu_PI, xseq)

text(x = mean(range(dcc$H)), y = max(dcc$W),
     labels = "Model: Height -> Weight", pos = 3, cex = 1.2)
```



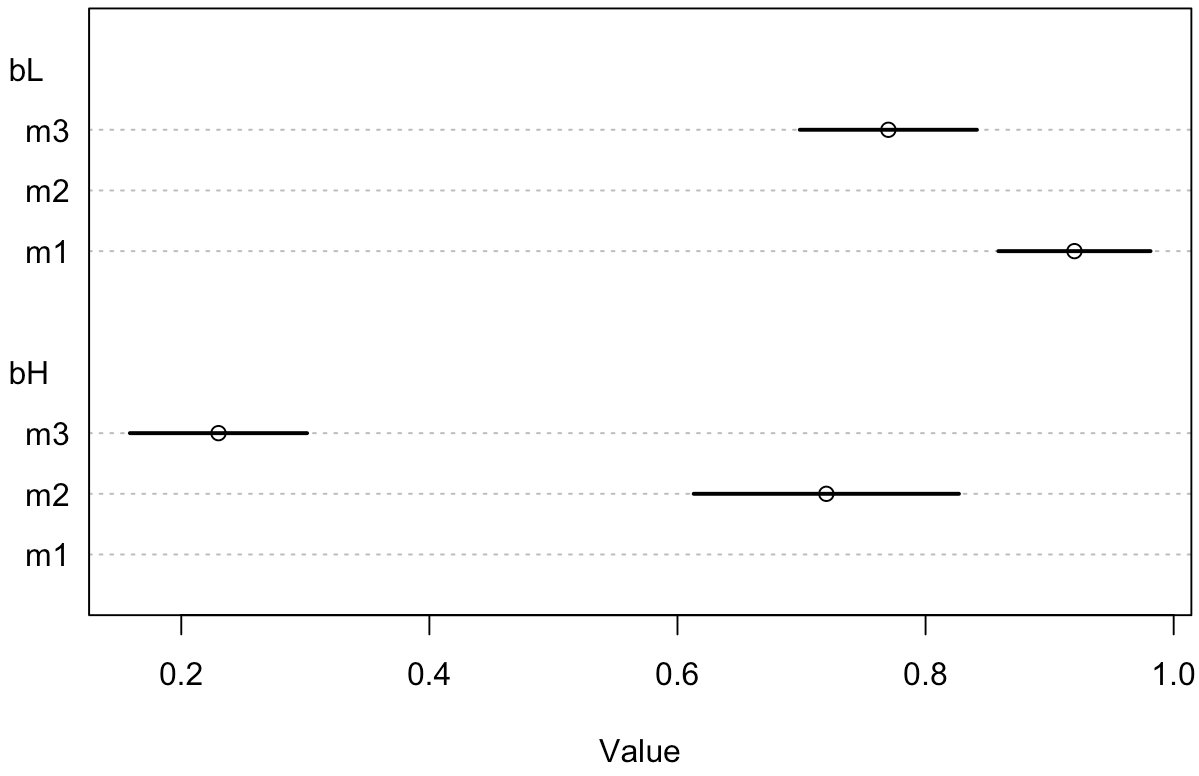
Model 3: Length 2 + Height -> Weight

```
m3 <- quap(
  alist(
    W ~ dnorm(mu, sigma),
    mu <- a + bL * L + bH * H,
    a ~ dnorm(0, 1),
```

```
    bL ~ dnorm(0.5, 2),
    bH ~ dnorm(0.5, 2),
    sigma ~ dexp(1)
  ), data = dcc
)
precis(m3)
```

```
              mean         sd        5.5%      94.5%
a     -1.066862e-07 0.02788313 -0.04456273 0.04456252
bL     7.707754e-01 0.03642616  0.71255935 0.82899142
bH     2.307623e-01 0.03642616  0.17254624 0.28897831
sigma  3.517296e-01 0.01969067  0.32026012 0.38319909
```

```
plot( coeftab( m1 , m2 , m3 ) , pars=c("bL","bH") )
```



Model 3 predicts **Weight (W)** using both **Length 2 (L)** and **Height (H)**. The prior for the intercept (**a**) is centered around 20 grams with a standard deviation of 5 grams, reflecting the baseline weight. The slope for **Length 2 (bL)** has a prior centered at 0 with a standard deviation of 1, and the slope for **Height (bH)** has a prior centered at 0.3 with a standard deviation of 0.2, suggesting a positive effect of height on weight.

M3: We see that 0.77 for the Length is still high but it decreased. The mean for Height also decreased significantly. going from 0.72 to 0.23. So when both x variables are used together their significance

decreases.

Comparison Plot

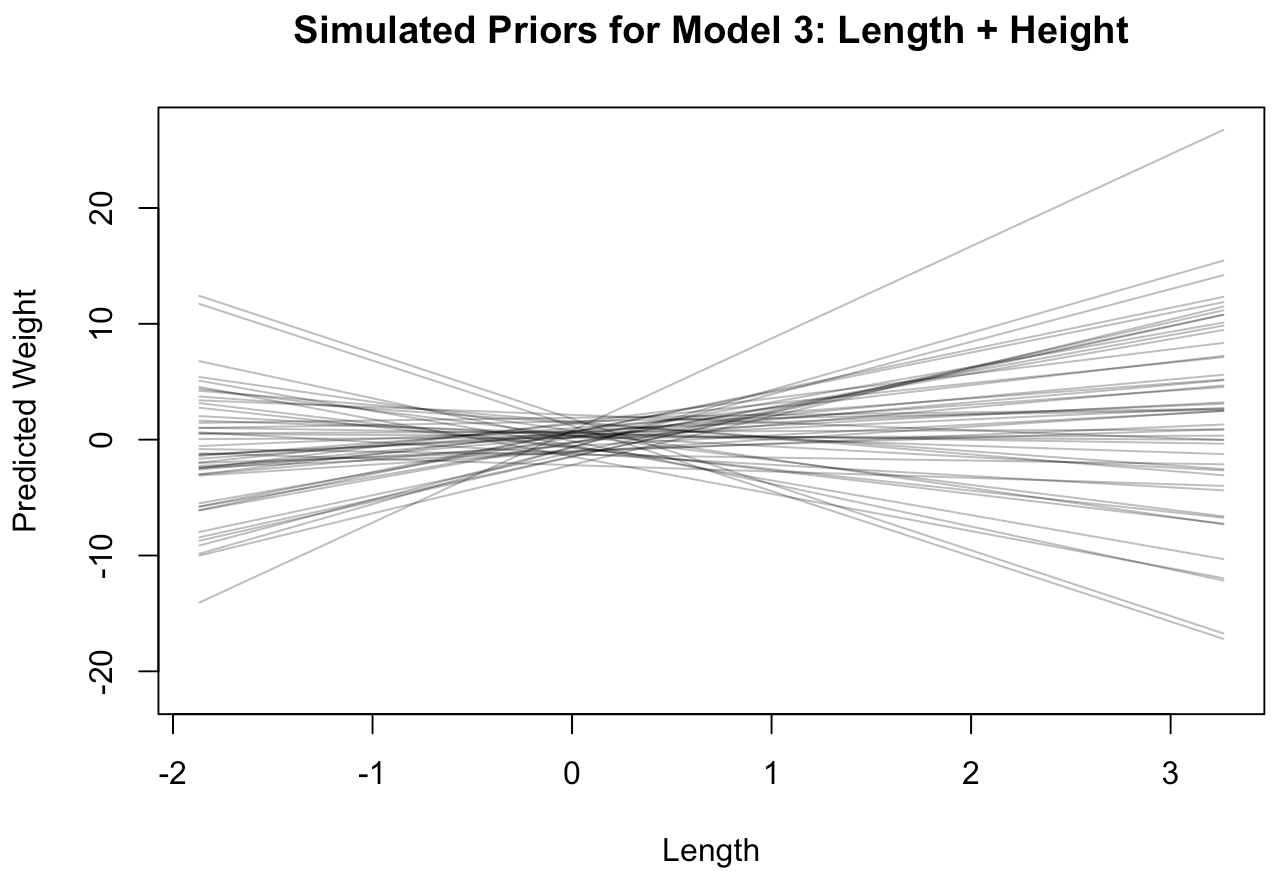
The circle means the average the lines how wide is the variability.

The plot shows that height (H) alone in M2 has high variability but a positive relationship with weight, indicating uncertainty in its predictive power. When combined with length (L) in M3, the variability decreases, but height's ability to predict weight is reduced. Length alone in M1 has low variability, showing it is a strong predictor of weight. However, combining length and height in M3 also reduces length's predictive strength. Both variables predict weight better individually than together.

- **Length (L)**:
  - Alone, it shows a **strong and consistent relationship** with weight, indicated by low variability in its estimate.
  - When combined with height, its predictive ability is slightly diminished, suggesting that height and length may overlap in the variance they explain for weight.
- **Height (H)**:
  - Alone, it has a **positive but more variable relationship** with weight, indicating greater uncertainty.
  - Combining it with length reduces its predictive power, likely due to overlap with length in explaining weight.
- **Joint Prediction**:
  - Both variables together do not predict weight as effectively as each variable individually. This suggests some redundancy, where both variables might explain similar aspects of weight variation.

Model 3: Simulate the priors

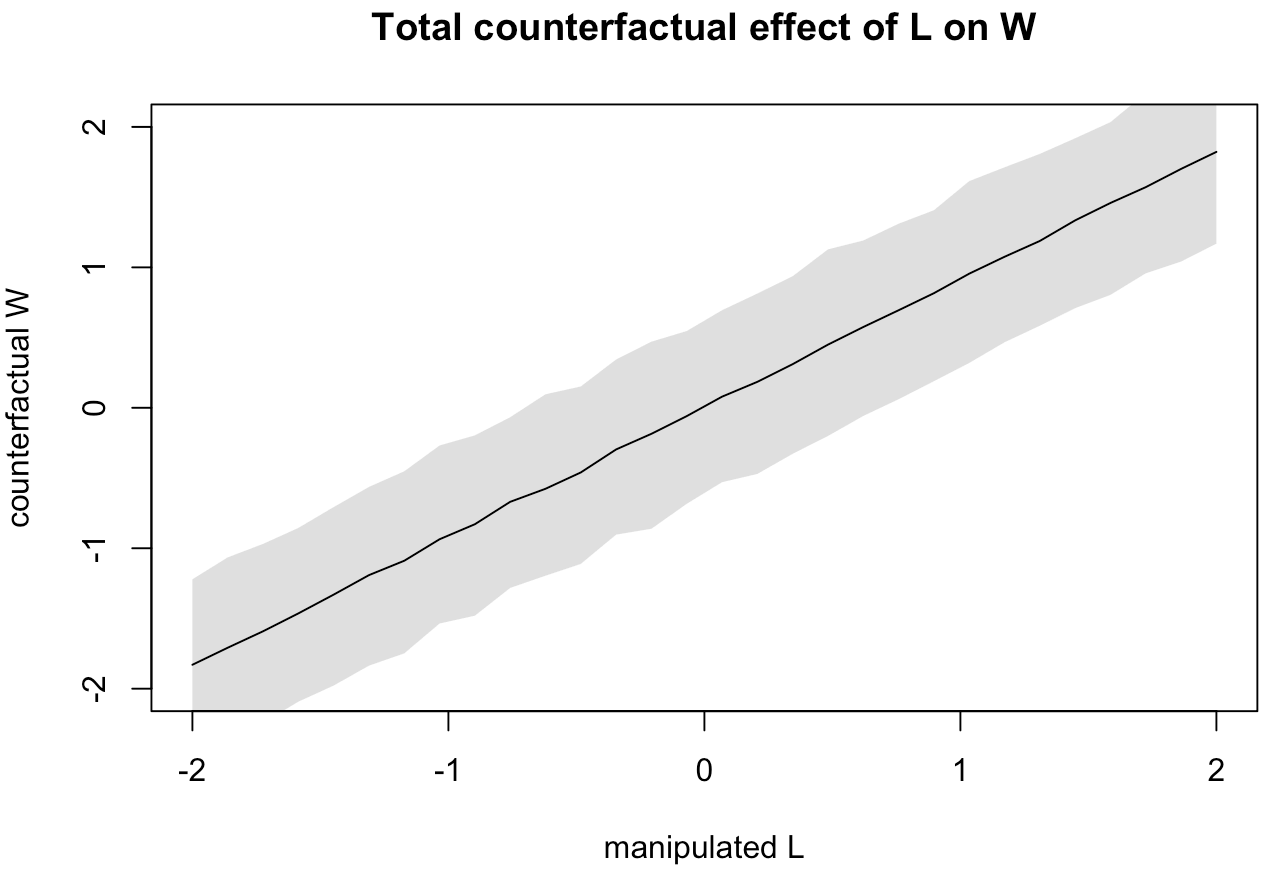
```
prior <- extract.prior(m3)

xseq_L <- seq(min(dcc$L), max(dcc$L), length.out = 100)

xseq_H <- seq(min(dcc$H), max(dcc$H), length.out = 100)

mu <- link(m3, post = prior, data = list(L = xseq_L, H = xseq_H))

plot(NULL, xlim = range(xseq_L), ylim = range(mu),
     xlab = "Length", ylab = "Predicted Weight", main = "Simulated Priors for Moc

for (i in 1:50) {
  lines(xseq_L, mu[i,], col = col.alpha("black", 0.3))
}
```



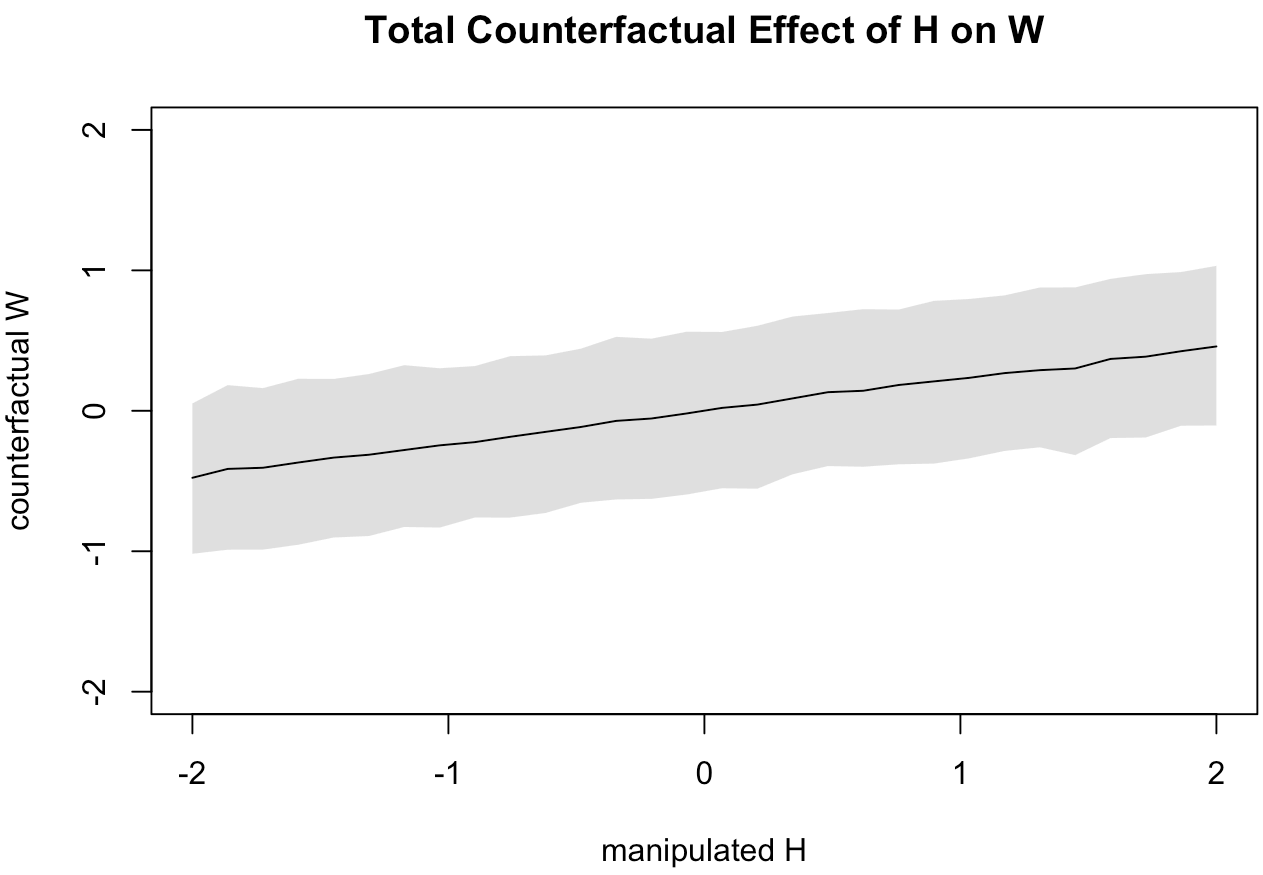
Counterfactual Plots for all Models

```
#counterfactual plots
m3.a <- quap(
  alist(
    ## L -> W <- H
    W ~ dnorm( mu , sigma ) ,
    mu <- a + bH*H + bL*L ,
    a ~ dnorm( 0 , 1 ) ,
    bH ~ dnorm( 0.5 , 2 ) ,
    bL ~ dnorm( 0.5 , 2 ) ,
    sigma ~ dexp( 1 ),
    ## L -> H
    H ~ dnorm( mu_H , sigma_H ),
    mu_H <- aH + bLH*L,
    aH ~ dnorm( 0 , 0.2 ),
    bLH ~ dnorm( 0 , 0.5 ),
    sigma_H ~ dexp( 1 )
    ) , data = dcc )

L_seq <- seq( from=-2 , to=2 , length.out=30 )

#L ON W
```

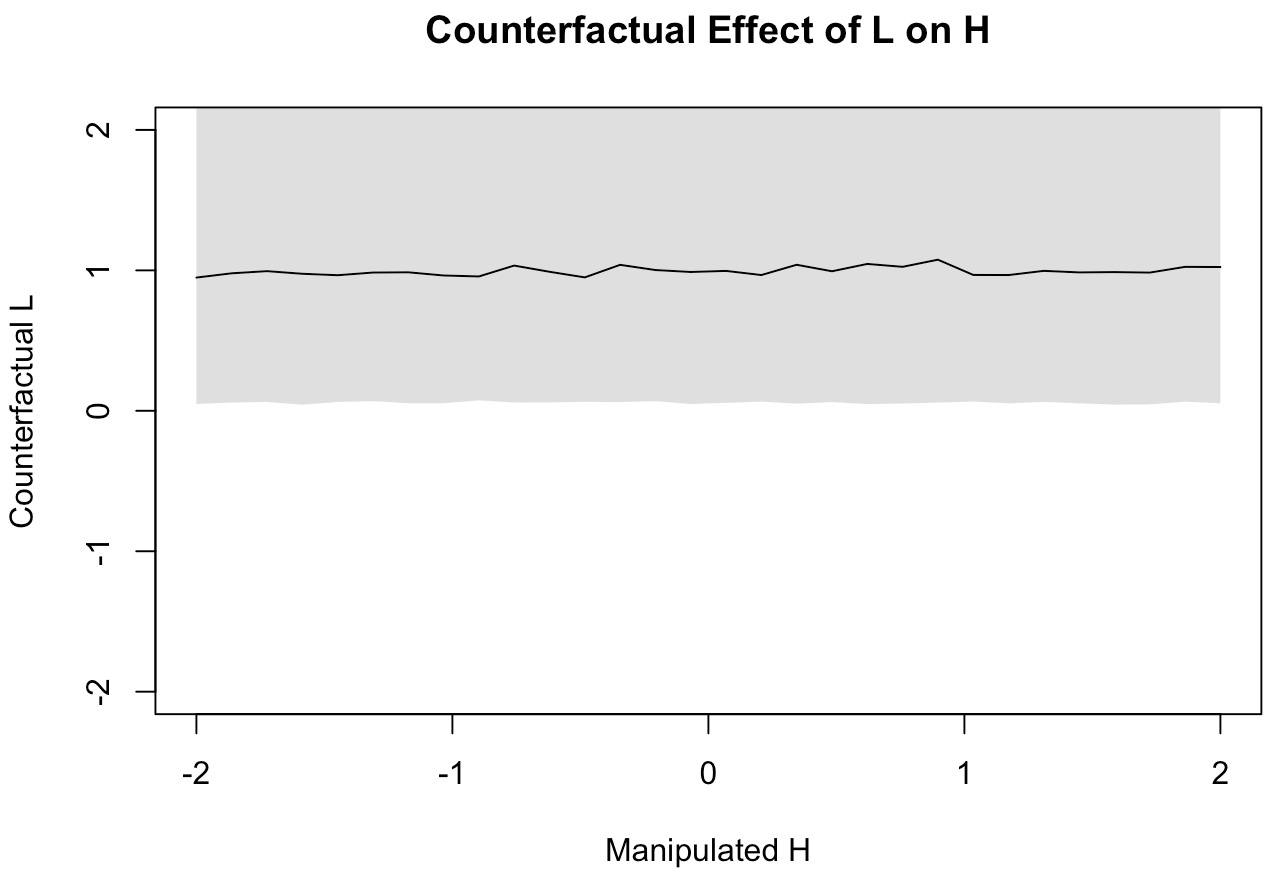
```
sim_dat <- data.frame( L=L_seq )

s <- sim( m3.a , data=sim_dat , vars=c("H","W") )


plot(sim_dat$L, colMeans(s$W), ylim = c(-2, 2), type = "l",
  main = "Total counterfactual effect of L on W",
  xlab = "manipulated L",
  ylab = "counterfactual W"
)

shade( apply(s$W,2,PI) , sim_dat$L )
```




```
#H ON W

sim_dat <- data.frame( H=seq(from=-2,to=2,length.out=30) , L=0 )

s <- sim( m3.a , data=sim_dat , vars=c("W") )

plot( sim_dat$H , colMeans(s) , ylim=c(-2,2) , type = "l", main = "Total Counterf

shade( apply(s,2,PI) , sim_dat$H )
```




```
#H ON L
sim_dat <- data.frame(H = seq(from = -2, to = 2, length.out = 30))
s <- sim(m3.a, data = sim_dat, vars = "L")  # Simulate L based on H

par(mar = c(5.1, 4.1, 4.1, 2.1))
plot(sim_dat$H, colMeans(s), ylim = c(-2, 2), type = "l",
     main = "Counterfactual Effect of L on H",
     xlab = "Manipulated H",
     ylab = "Counterfactual L")

shade(apply(s, 2, PI), sim_dat$H)
```



1. Length does good job at explaining weight -2 to 2
2. Height is OK/decent -1 to 0 has a decent relation ship
3. Not a really significant relation, because its at 1 the relationship exist but it doesnt mean anythign

Markov Chain Monte Carlo

MCMC - Model 1: Length2 -> Weight

```
# MCMC - Model 1: Length2 -> Weight
m1_mcmc <- ulam(
  alist(
    W ~ dnorm(mu, sigma),
    mu <- a + bL * L,
    a ~ dnorm(0, 1),
    bL ~ dnorm(0.5, 2),
    sigma ~ dexp(1)
  ), data = dcc,
  chains = 4,
  cores = 4,
  log_lik = TRUE
)
```

```
Removing one or more character or factor variables:

Species

Running MCMC with 4 parallel chains, with 1 thread(s) per chain...

Chain 1 Iteration:   1 / 1000 [  0%]  (Warmup)
Chain 1 Iteration: 100 / 1000 [ 10%]  (Warmup)
Chain 1 Iteration: 200 / 1000 [ 20%]  (Warmup)
Chain 1 Iteration: 300 / 1000 [ 30%]  (Warmup)
Chain 1 Iteration: 400 / 1000 [ 40%]  (Warmup)
Chain 1 Iteration: 500 / 1000 [ 50%]  (Warmup)
Chain 1 Iteration: 501 / 1000 [ 50%]  (Sampling)
Chain 1 Iteration: 600 / 1000 [ 60%]  (Sampling)
Chain 1 Iteration: 700 / 1000 [ 70%]  (Sampling)
Chain 1 Iteration: 800 / 1000 [ 80%]  (Sampling)
Chain 1 Iteration: 900 / 1000 [ 90%]  (Sampling)
Chain 1 Iteration: 1000 / 1000 [100%]  (Sampling)
Chain 2 Iteration:   1 / 1000 [  0%]  (Warmup)
Chain 2 Iteration: 100 / 1000 [ 10%]  (Warmup)
Chain 2 Iteration: 200 / 1000 [ 20%]  (Warmup)
Chain 2 Iteration: 300 / 1000 [ 30%]  (Warmup)
Chain 2 Iteration: 400 / 1000 [ 40%]  (Warmup)
Chain 2 Iteration: 500 / 1000 [ 50%]  (Warmup)
Chain 2 Iteration: 501 / 1000 [ 50%]  (Sampling)
Chain 2 Iteration: 600 / 1000 [ 60%]  (Sampling)
Chain 2 Iteration: 700 / 1000 [ 70%]  (Sampling)
Chain 2 Iteration: 800 / 1000 [ 80%]  (Sampling)
Chain 2 Iteration: 900 / 1000 [ 90%]  (Sampling)
Chain 2 Iteration: 1000 / 1000 [100%]  (Sampling)
Chain 3 Iteration:   1 / 1000 [  0%]  (Warmup)
Chain 3 Iteration: 100 / 1000 [ 10%]  (Warmup)
Chain 3 Iteration: 200 / 1000 [ 20%]  (Warmup)
Chain 3 Iteration: 300 / 1000 [ 30%]  (Warmup)
Chain 3 Iteration: 400 / 1000 [ 40%]  (Warmup)
Chain 3 Iteration: 500 / 1000 [ 50%]  (Warmup)
Chain 3 Iteration: 501 / 1000 [ 50%]  (Sampling)
Chain 3 Iteration: 600 / 1000 [ 60%]  (Sampling)
Chain 3 Iteration: 700 / 1000 [ 70%]  (Sampling)
Chain 3 Iteration: 800 / 1000 [ 80%]  (Sampling)
Chain 3 Iteration: 900 / 1000 [ 90%]  (Sampling)
Chain 3 Iteration: 1000 / 1000 [100%]  (Sampling)
Chain 4 Iteration:   1 / 1000 [  0%]  (Warmup)
Chain 4 Iteration: 100 / 1000 [ 10%]  (Warmup)
Chain 4 Iteration: 200 / 1000 [ 20%]  (Warmup)
Chain 4 Iteration: 300 / 1000 [ 30%]  (Warmup)
Chain 4 Iteration: 400 / 1000 [ 40%]  (Warmup)
Chain 4 Iteration: 500 / 1000 [ 50%]  (Warmup)
Chain 4 Iteration: 501 / 1000 [ 50%]  (Sampling)
Chain 4 Iteration: 600 / 1000 [ 60%]  (Sampling)
Chain 4 Iteration: 700 / 1000 [ 70%]  (Sampling)
```

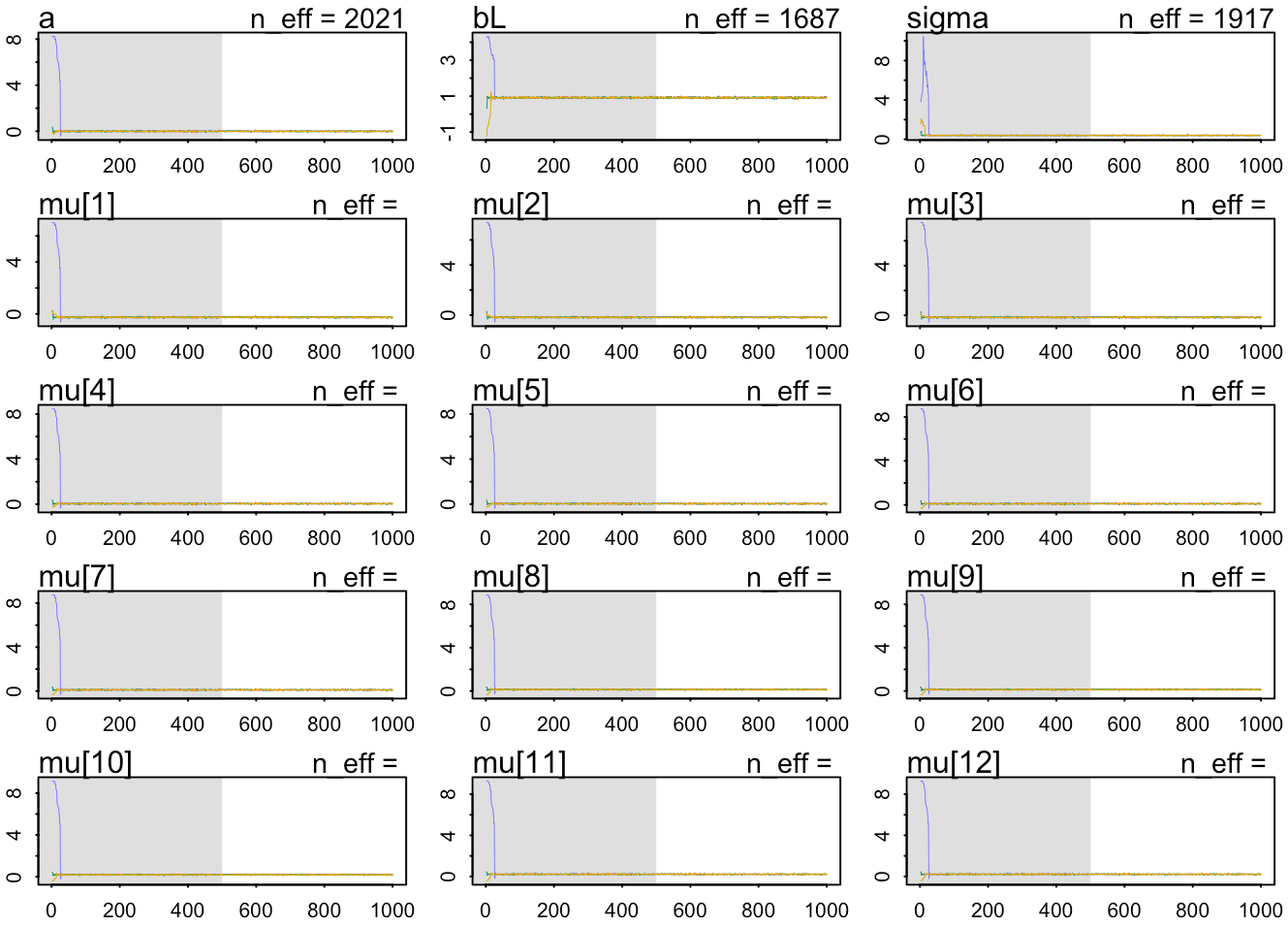
```
Chain 4 Iteration: 800 / 1000 [ 80%]  (Sampling)
Chain 4 Iteration: 900 / 1000 [ 90%]  (Sampling)
Chain 4 Iteration: 1000 / 1000 [100%]  (Sampling)
Chain 1 finished in 0.1 seconds.
Chain 2 finished in 0.1 seconds.
Chain 3 finished in 0.1 seconds.
Chain 4 finished in 0.1 seconds.

All 4 chains finished successfully.
Mean chain execution time: 0.1 seconds.
Total execution time: 0.2 seconds.
```

```
precis(m1_mcmc)
```

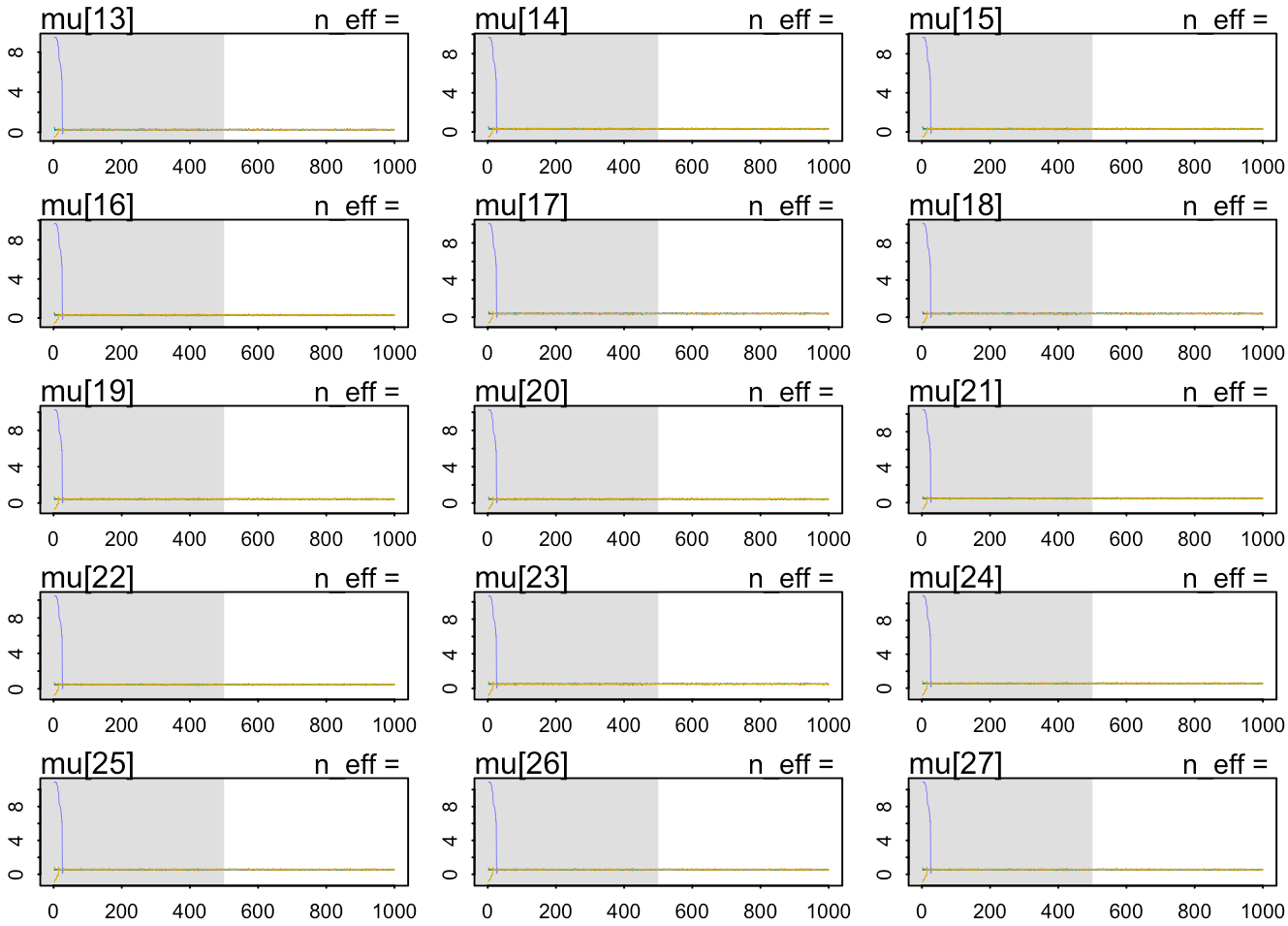
```
               mean         sd        5.5%      94.5%     rhat ess_bulk
a     -0.0002391809 0.03239622 -0.05260939 0.05015526 1.001384 2021.163
bL     0.9176218820 0.03193506  0.86700614 0.96993348 1.002102 1686.831
sigma  0.3990582270 0.02367038  0.36317978 0.43783203 1.000609 1917.424
```

```
traceplot(m1_mcmc)
```

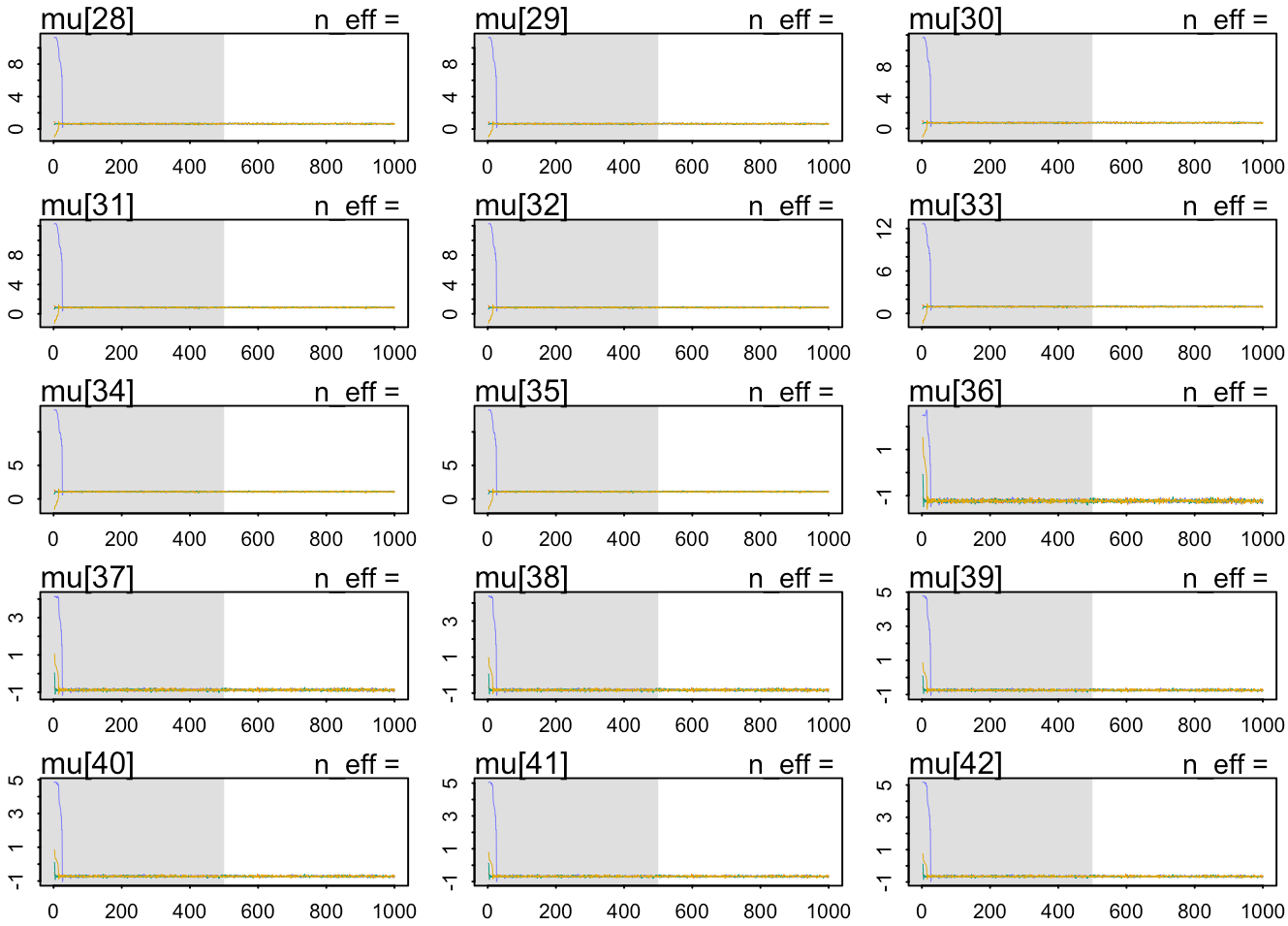
```
Waiting to draw page 2 of 11
```



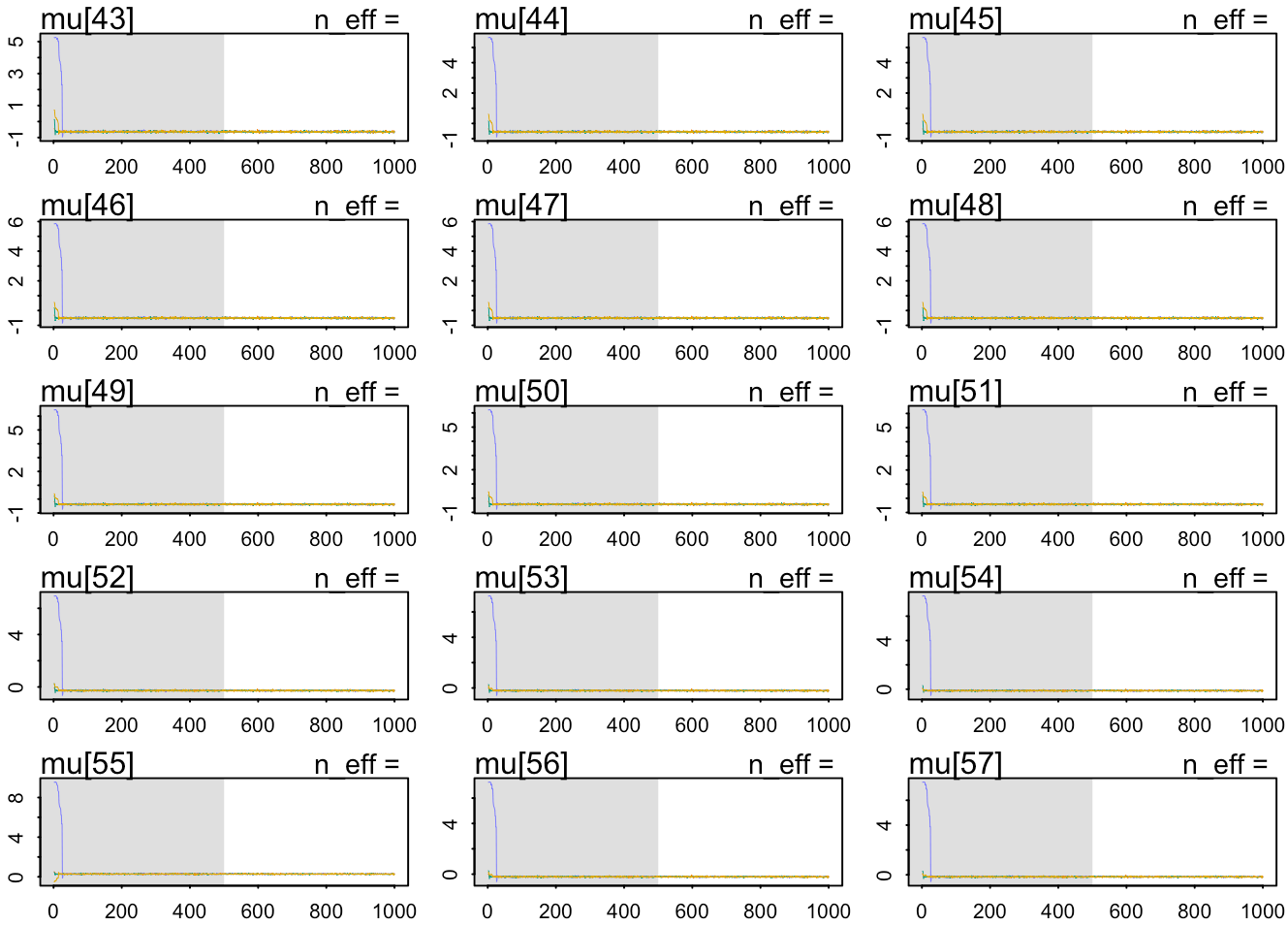
```
Waiting to draw page 3 of 11
```



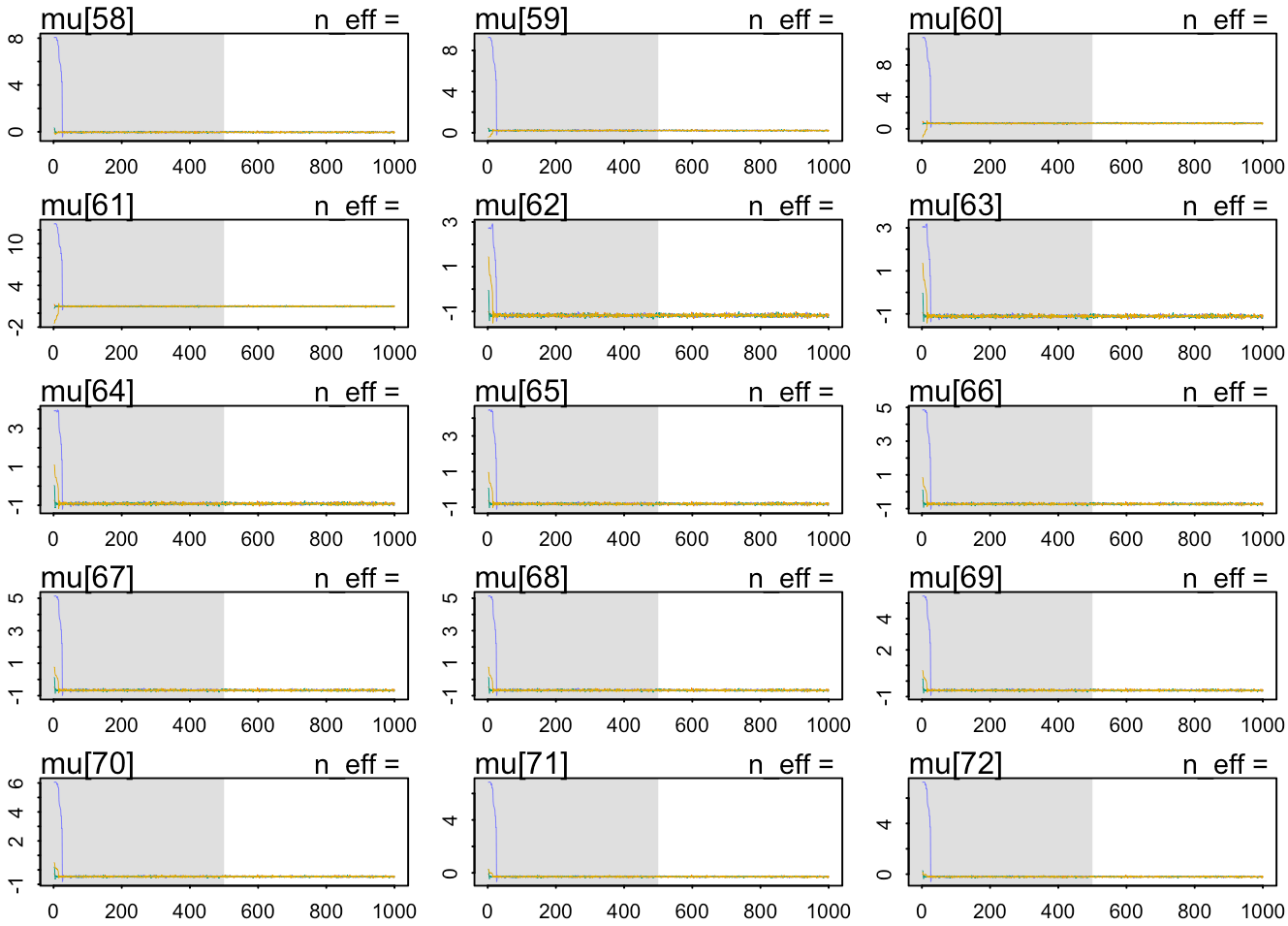

```
Waiting to draw page 4 of 11
```



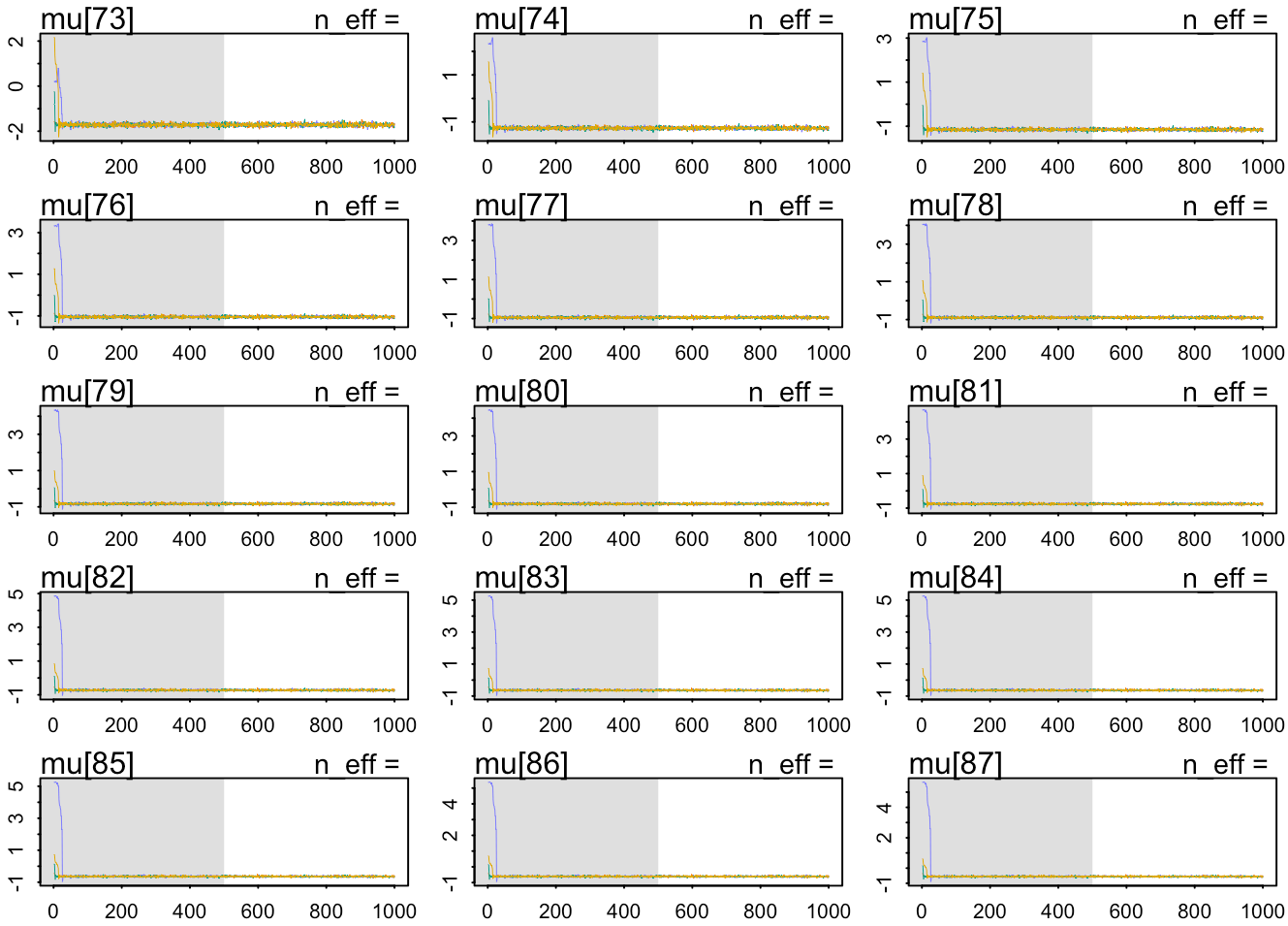
Waiting to draw page 5 of 11



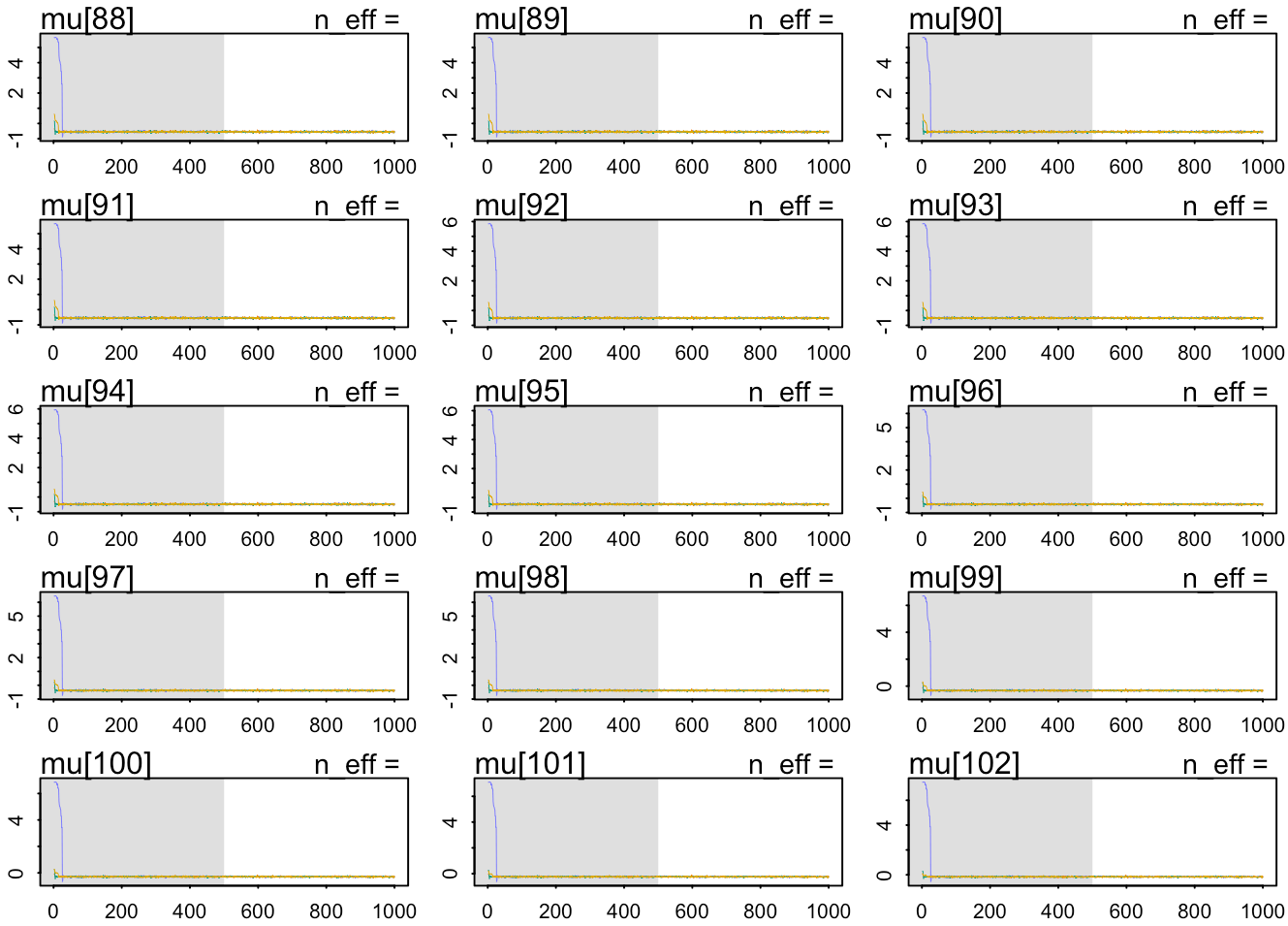
Waiting to draw page 6 of 11



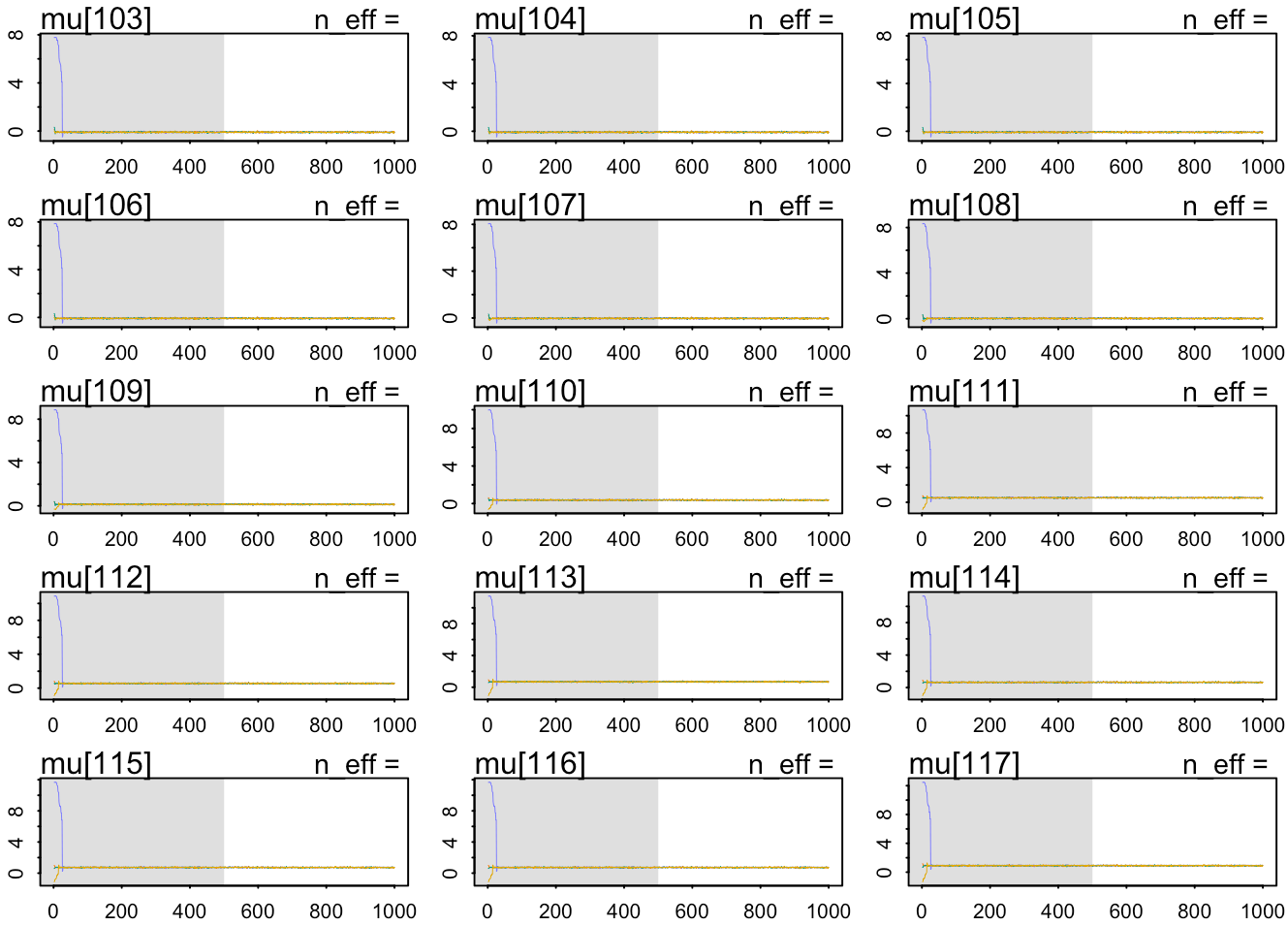
Waiting to draw page 7 of 11



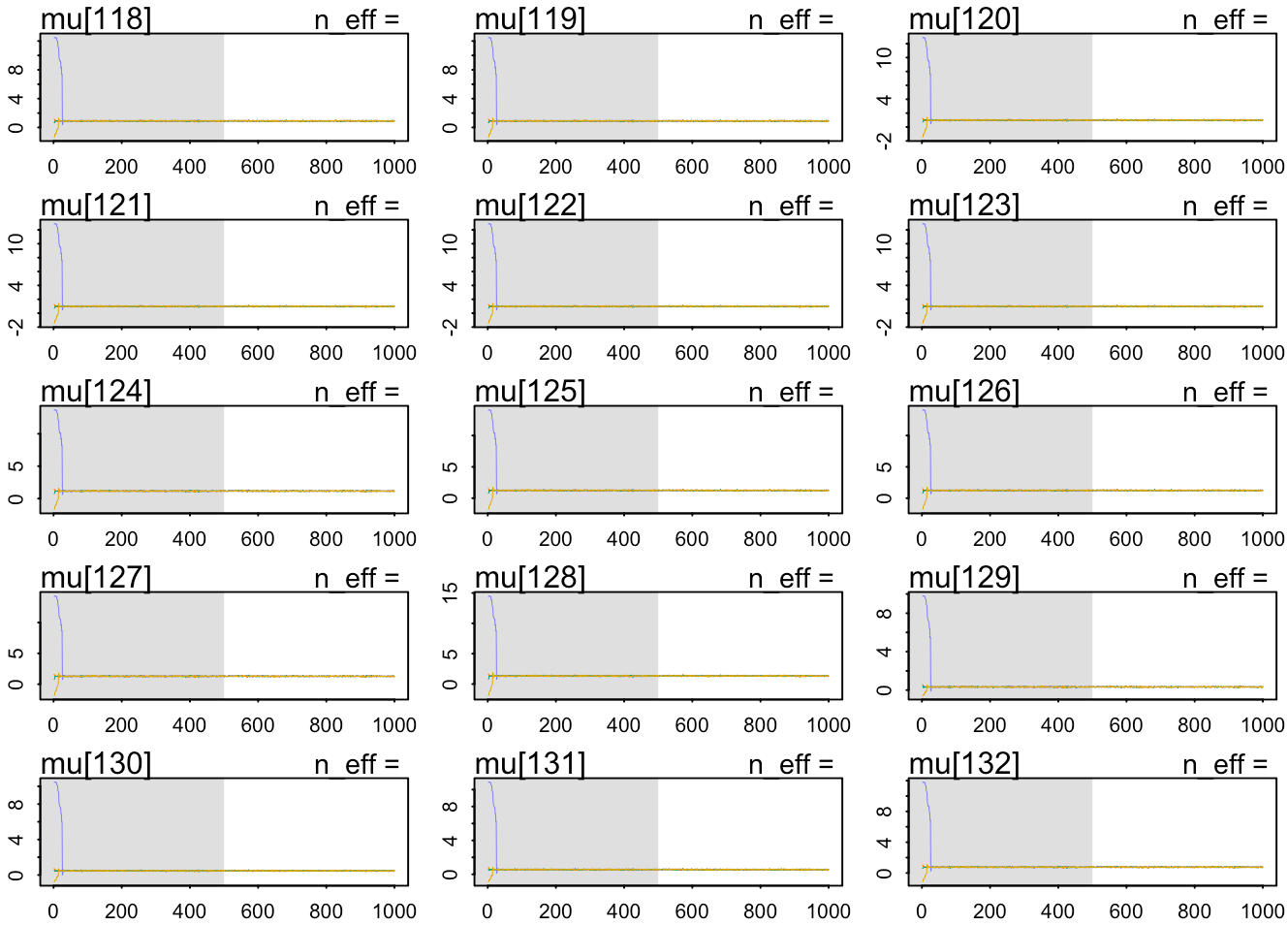
Waiting to draw page 8 of 11



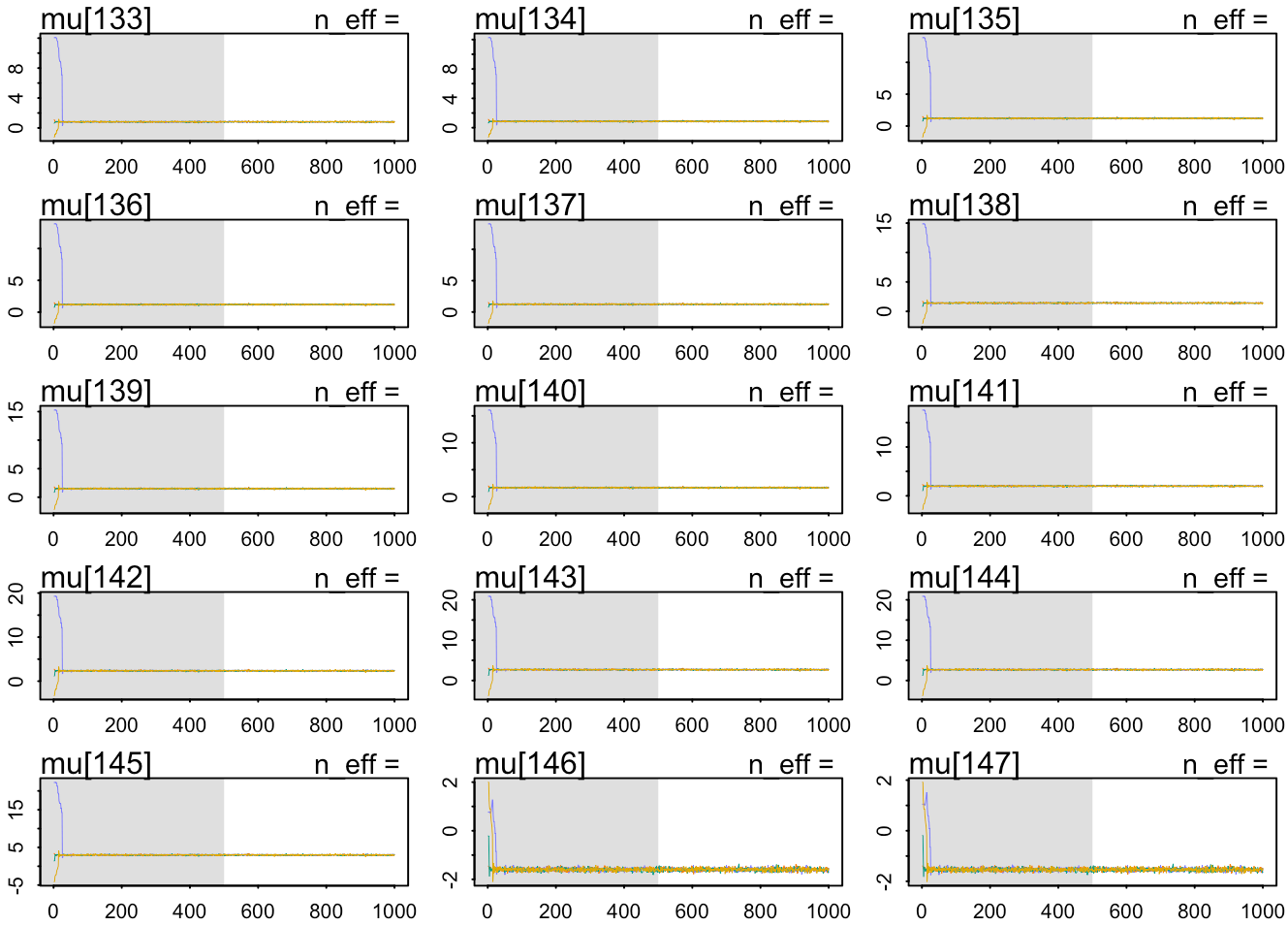
Waiting to draw page 9 of 11



Waiting to draw page 10 of 11

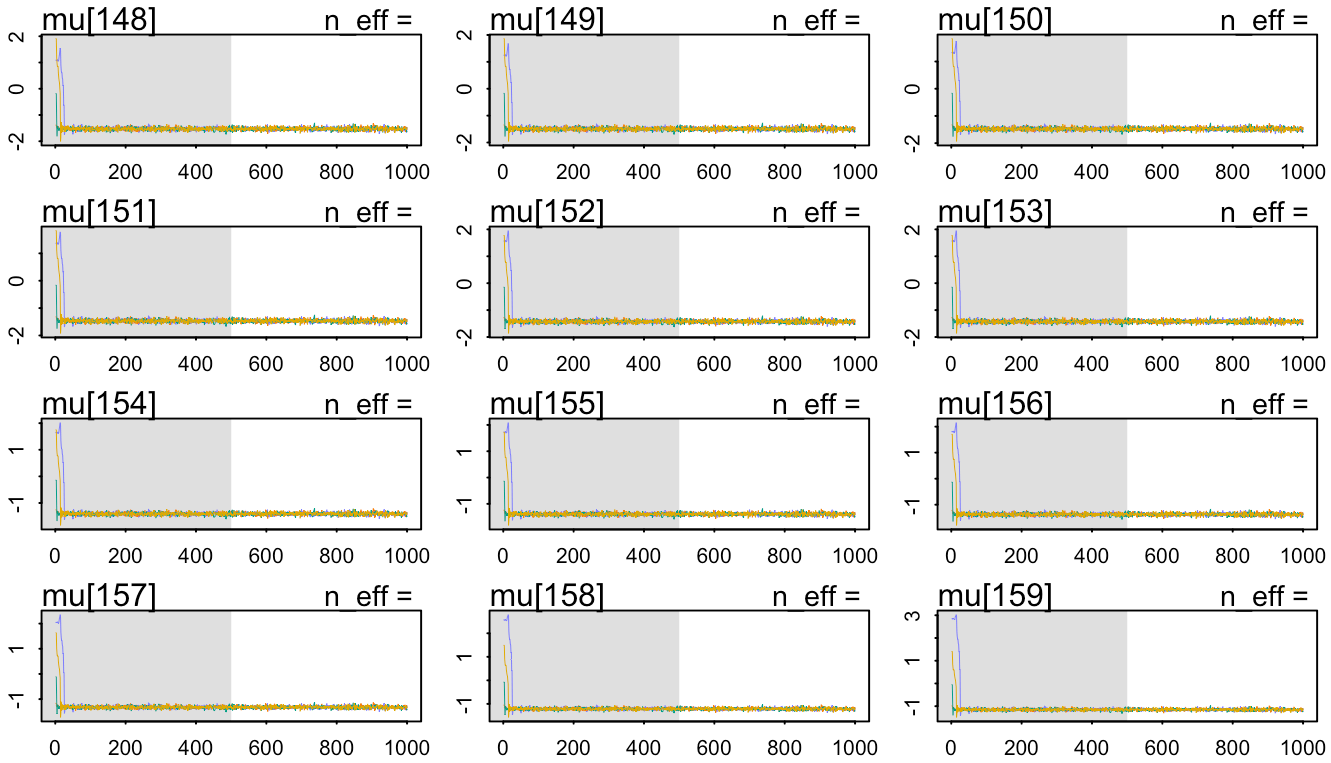


Waiting to draw page 11 of 11




```
WAIC(m1_mcmc)
```

```
      WAIC      lppd  penalty  std_err
1 162.2902 -77.15351 3.991606 20.43395
```



MCMC - Model 2: Height -> Weight

```
# MCMC - Model 2: Height -> Weight
m2_mcmc <- ulam(
  alist(
    W ~ dnorm(mu, sigma),
    mu <- a + bH * H,
    a ~ dnorm(0, 1),
    bH ~ dnorm(0.5, 2),
    sigma ~ dexp(1)
  ), data = dcc,
  chains = 4,
  cores = 4,
  log_lik = TRUE
)
```

```
Removing one or more character or factor variables:

Species

Running MCMC with 4 parallel chains, with 1 thread(s) per chain...

Chain 1 Iteration:   1 / 1000 [  0%]  (Warmup)
Chain 1 Iteration: 100 / 1000 [ 10%]  (Warmup)
```

```
Chain 1 Iteration: 200 / 1000 [ 20%]  (Warmup)
Chain 1 Iteration: 300 / 1000 [ 30%]  (Warmup)
Chain 1 Iteration: 400 / 1000 [ 40%]  (Warmup)
Chain 1 Iteration: 500 / 1000 [ 50%]  (Warmup)
Chain 1 Iteration: 501 / 1000 [ 50%]  (Sampling)
Chain 1 Iteration: 600 / 1000 [ 60%]  (Sampling)
Chain 1 Iteration: 700 / 1000 [ 70%]  (Sampling)
Chain 1 Iteration: 800 / 1000 [ 80%]  (Sampling)
Chain 1 Iteration: 900 / 1000 [ 90%]  (Sampling)
Chain 1 Iteration: 1000 / 1000 [100%]  (Sampling)
Chain 2 Iteration:   1 / 1000 [  0%]  (Warmup)
Chain 2 Iteration: 100 / 1000 [ 10%]  (Warmup)
Chain 2 Iteration: 200 / 1000 [ 20%]  (Warmup)
Chain 2 Iteration: 300 / 1000 [ 30%]  (Warmup)
Chain 2 Iteration: 400 / 1000 [ 40%]  (Warmup)
Chain 2 Iteration: 500 / 1000 [ 50%]  (Warmup)
Chain 2 Iteration: 501 / 1000 [ 50%]  (Sampling)
Chain 2 Iteration: 600 / 1000 [ 60%]  (Sampling)
Chain 2 Iteration: 700 / 1000 [ 70%]  (Sampling)
Chain 2 Iteration: 800 / 1000 [ 80%]  (Sampling)
Chain 2 Iteration: 900 / 1000 [ 90%]  (Sampling)
Chain 2 Iteration: 1000 / 1000 [100%]  (Sampling)
Chain 2 Informational Message: The current Metropolis proposal is about to be rejected
because of the following issue:

Chain 2 Exception: normal_lpdf: Scale parameter is 0, but must be positive! (in
'/var/folders/ym/vpwglz617b78sp1lhzq6ydx00000gn/T/RtmpJMlyzU/model-3e9d24418835.stan',
line 25, column 4 to column 29)

Chain 2 If this warning occurs sporadically, such as for highly constrained variable
types like covariance matrices, then the sampler is fine,

Chain 2 but if this warning occurs often then your model may be either severely ill-
conditioned or misspecified.

Chain 2

Chain 3 Iteration:   1 / 1000 [  0%]  (Warmup)
Chain 3 Iteration: 100 / 1000 [ 10%]  (Warmup)
Chain 3 Iteration: 200 / 1000 [ 20%]  (Warmup)
Chain 3 Iteration: 300 / 1000 [ 30%]  (Warmup)
Chain 3 Iteration: 400 / 1000 [ 40%]  (Warmup)
Chain 3 Iteration: 500 / 1000 [ 50%]  (Warmup)
Chain 3 Iteration: 501 / 1000 [ 50%]  (Sampling)
Chain 3 Iteration: 600 / 1000 [ 60%]  (Sampling)
Chain 3 Iteration: 700 / 1000 [ 70%]  (Sampling)
Chain 3 Iteration: 800 / 1000 [ 80%]  (Sampling)
Chain 3 Iteration: 900 / 1000 [ 90%]  (Sampling)
Chain 3 Iteration: 1000 / 1000 [100%]  (Sampling)
Chain 4 Iteration:   1 / 1000 [  0%]  (Warmup)
Chain 4 Iteration: 100 / 1000 [ 10%]  (Warmup)
```

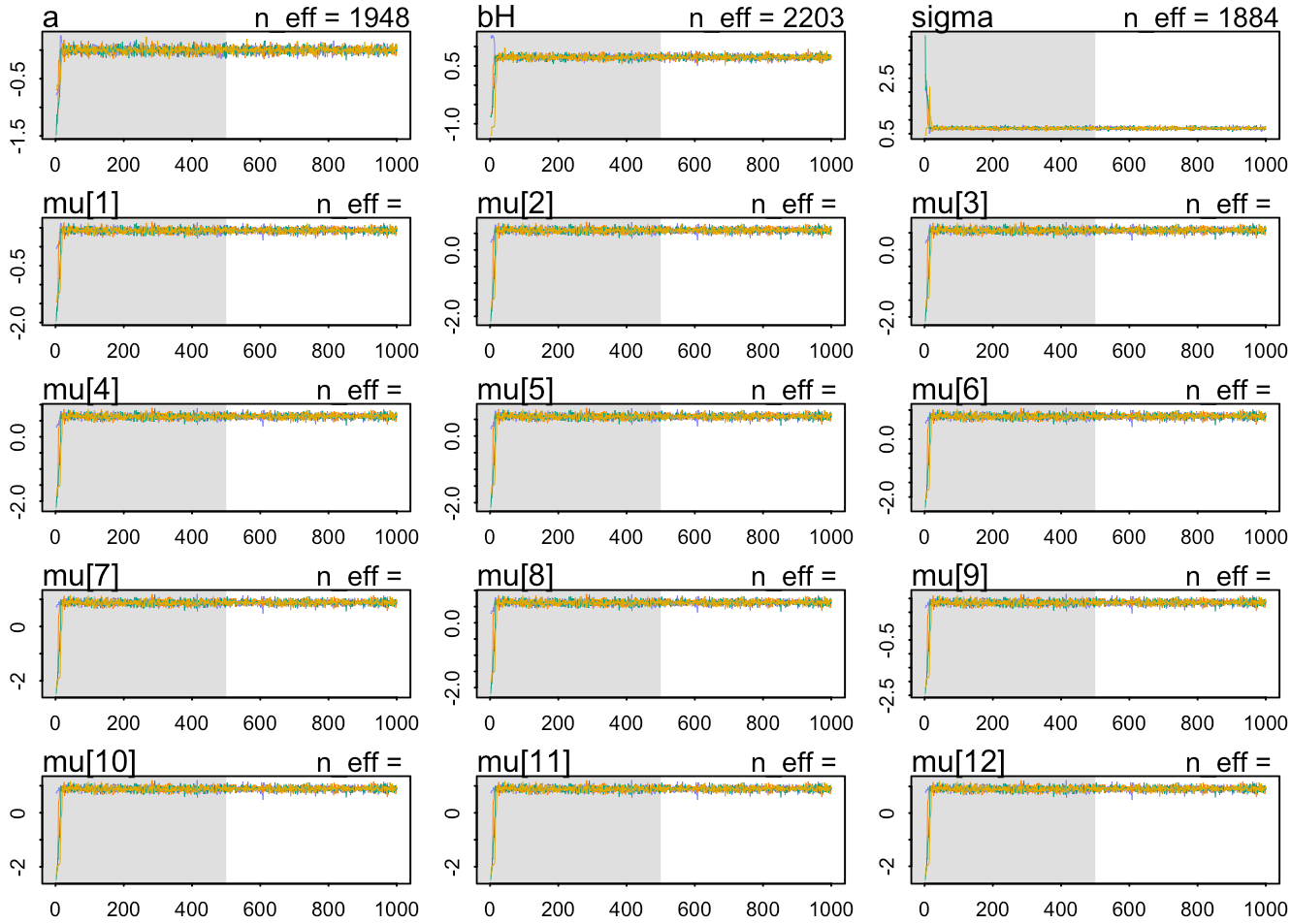
```
Chain 4 Iteration: 200 / 1000 [ 20%]  (Warmup)
Chain 4 Iteration: 300 / 1000 [ 30%]  (Warmup)
Chain 4 Iteration: 400 / 1000 [ 40%]  (Warmup)
Chain 4 Iteration: 500 / 1000 [ 50%]  (Warmup)
Chain 4 Iteration: 501 / 1000 [ 50%]  (Sampling)
Chain 4 Iteration: 600 / 1000 [ 60%]  (Sampling)
Chain 4 Iteration: 700 / 1000 [ 70%]  (Sampling)
Chain 4 Iteration: 800 / 1000 [ 80%]  (Sampling)
Chain 4 Iteration: 900 / 1000 [ 90%]  (Sampling)
Chain 4 Iteration: 1000 / 1000 [100%]  (Sampling)
Chain 1 finished in 0.1 seconds.
Chain 2 finished in 0.1 seconds.
Chain 3 finished in 0.1 seconds.
Chain 4 finished in 0.1 seconds.

All 4 chains finished successfully.
Mean chain execution time: 0.1 seconds.
Total execution time: 0.2 seconds.
```

```
precis(m2_mcmc)
```

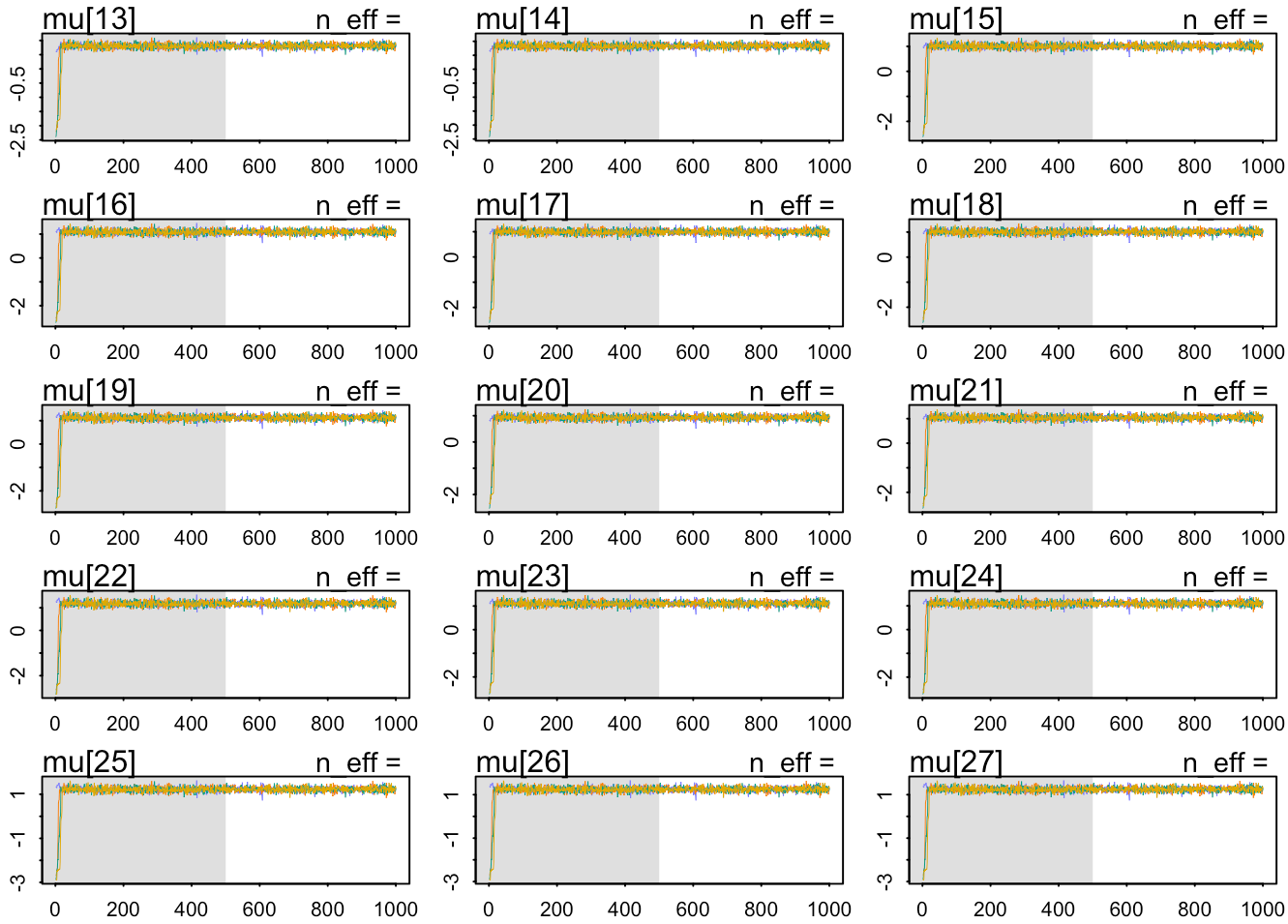
```
              mean         sd        5.5%     94.5%      rhat ess_bulk
a     0.0004709704 0.05272183 -0.08275567 0.0855989 0.9996331 1947.515
bH    0.7233994790 0.05601124  0.63704289 0.8108922 0.9997452 2202.755
sigma 0.6943400920 0.03967230  0.63460187 0.7592110 1.0026747 1883.757
```

```
traceplot(m2_mcmc)
```

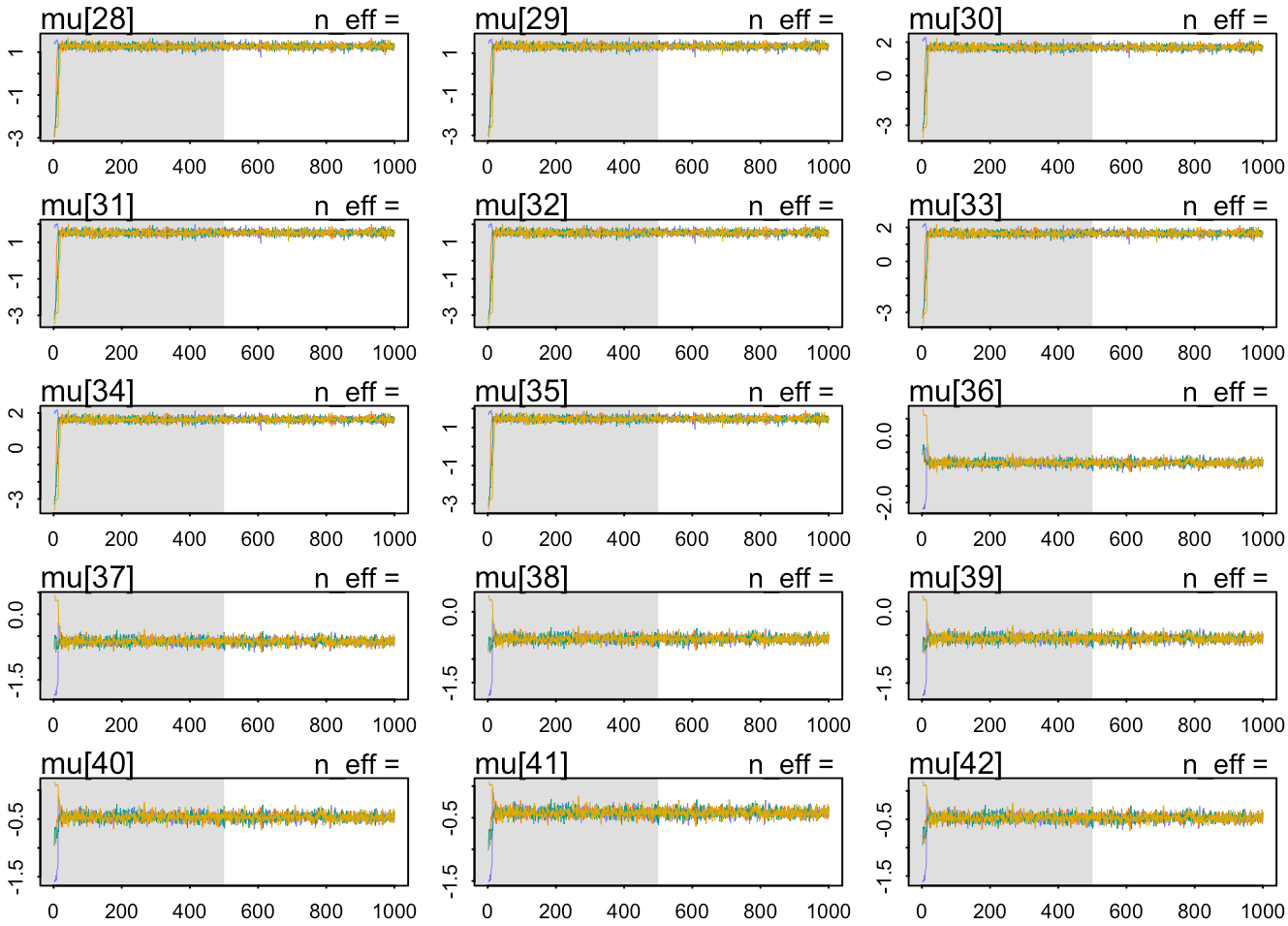
```
Waiting to draw page 2 of 11
```



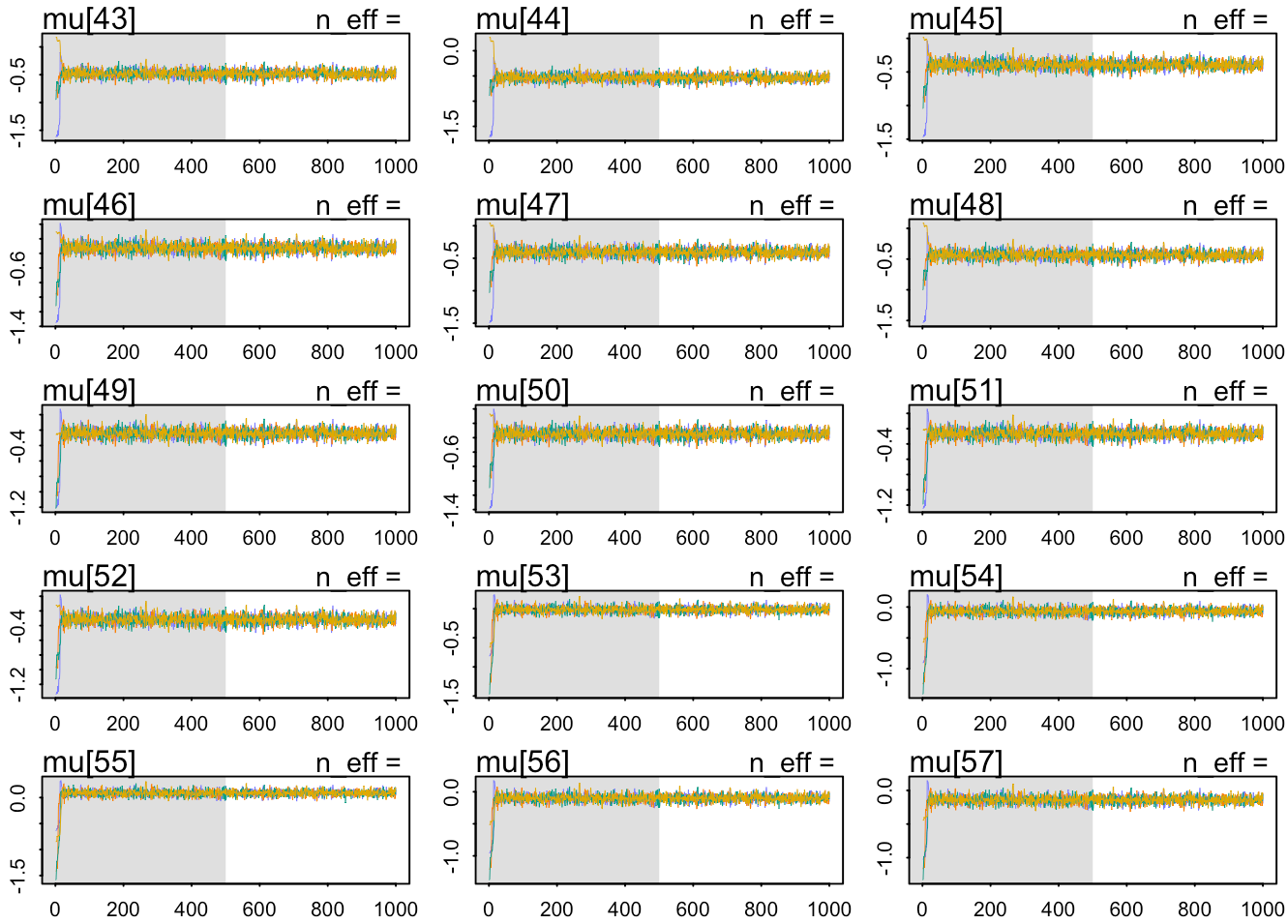
Waiting to draw page 3 of 11



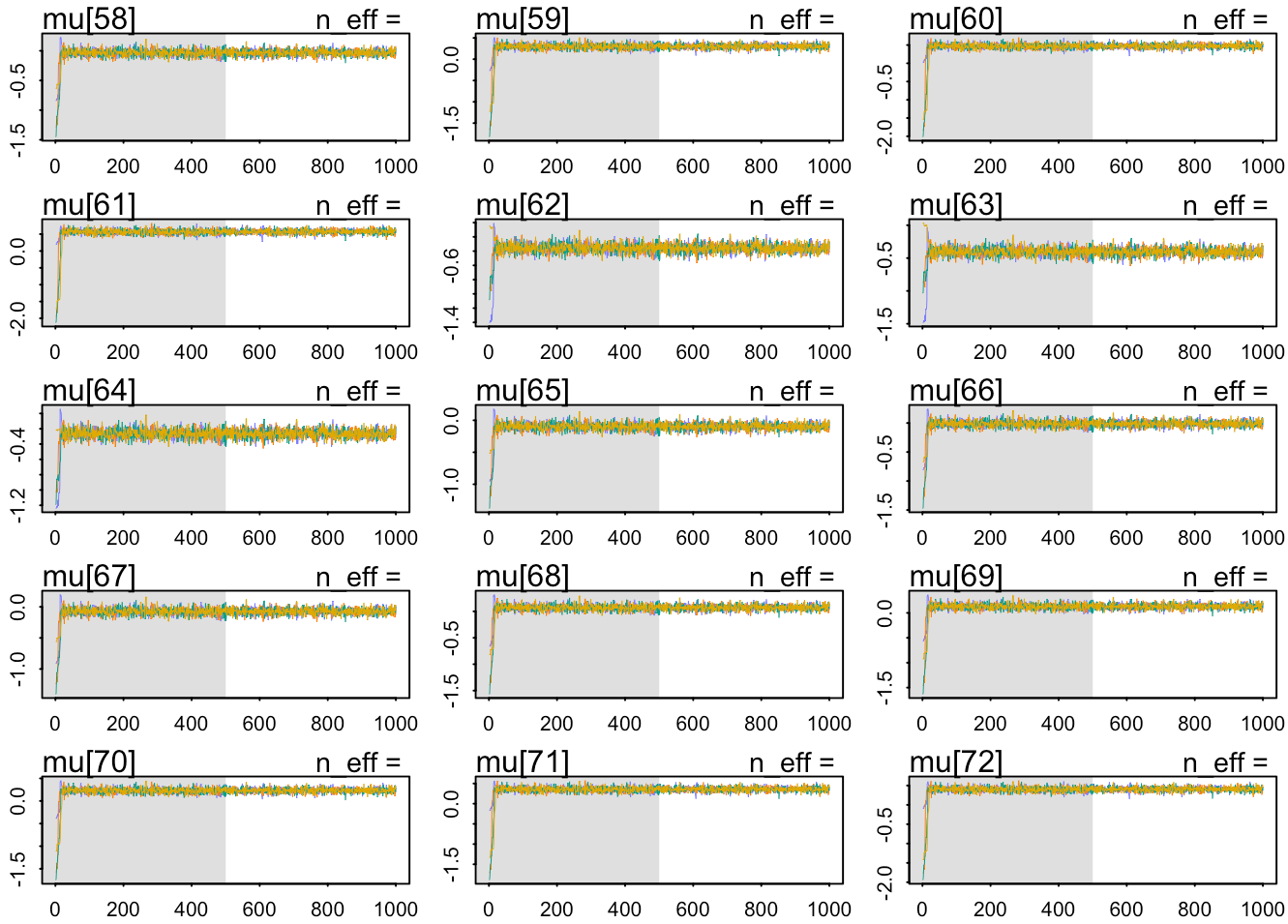
Waiting to draw page 4 of 11



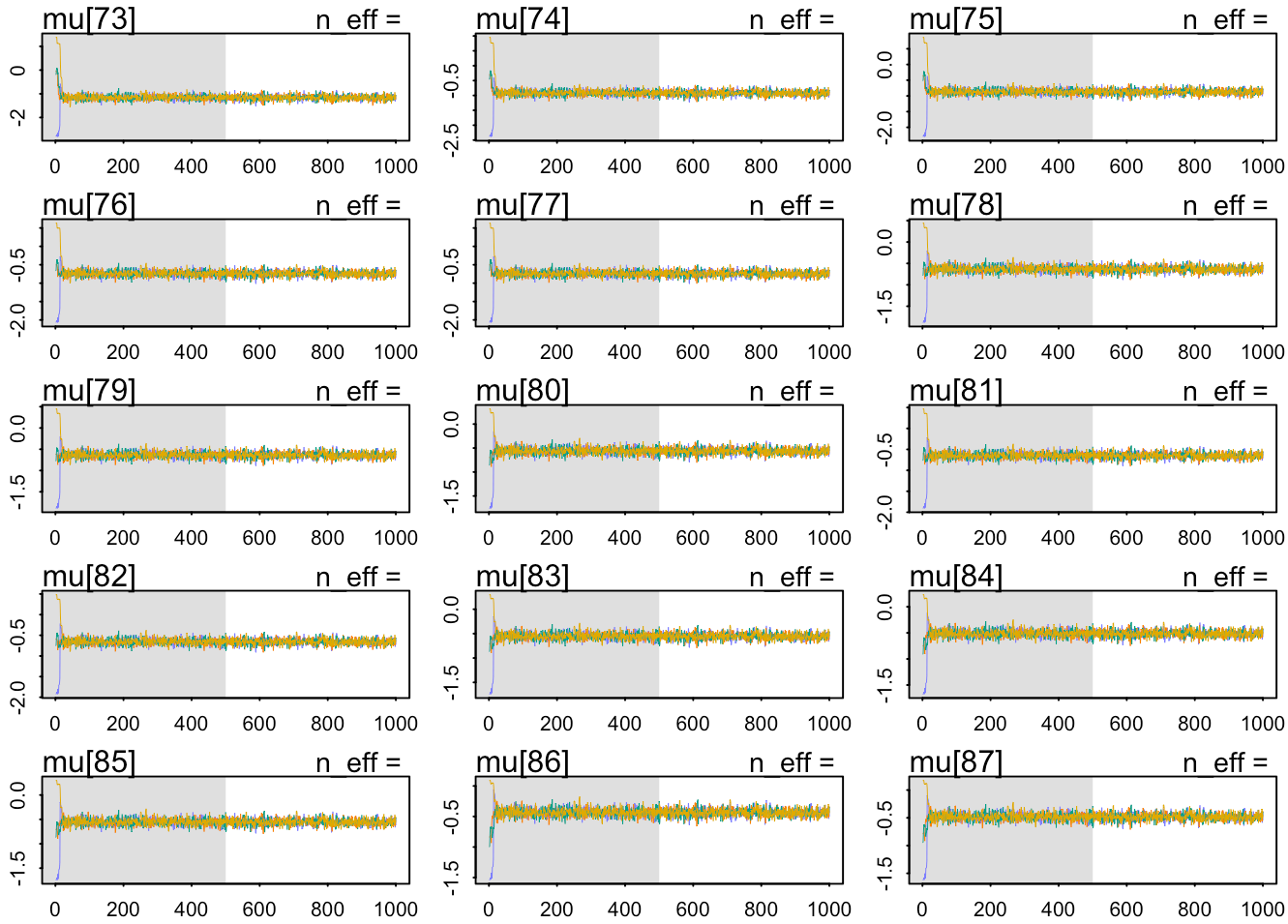
Waiting to draw page 5 of 11



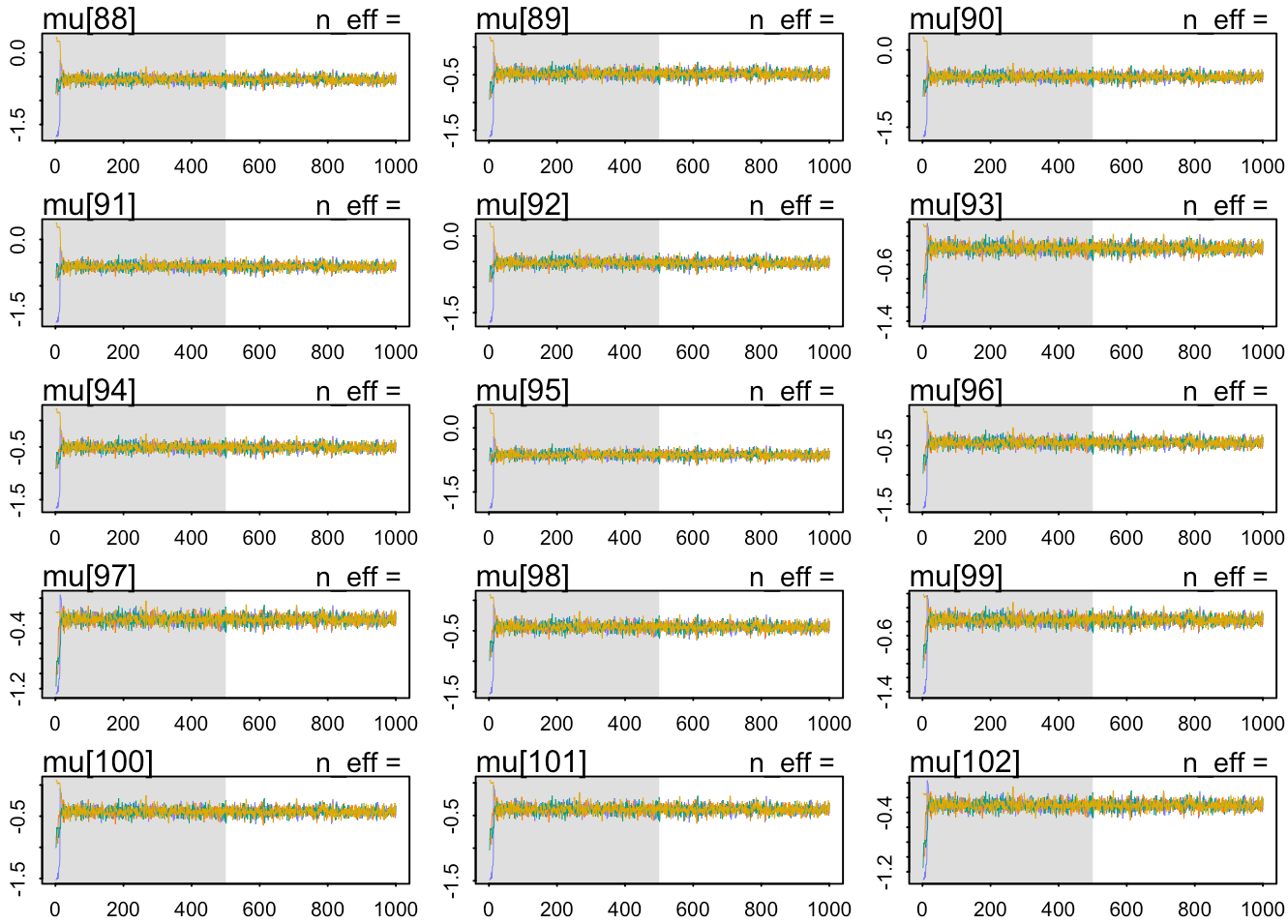
Waiting to draw page 6 of 11



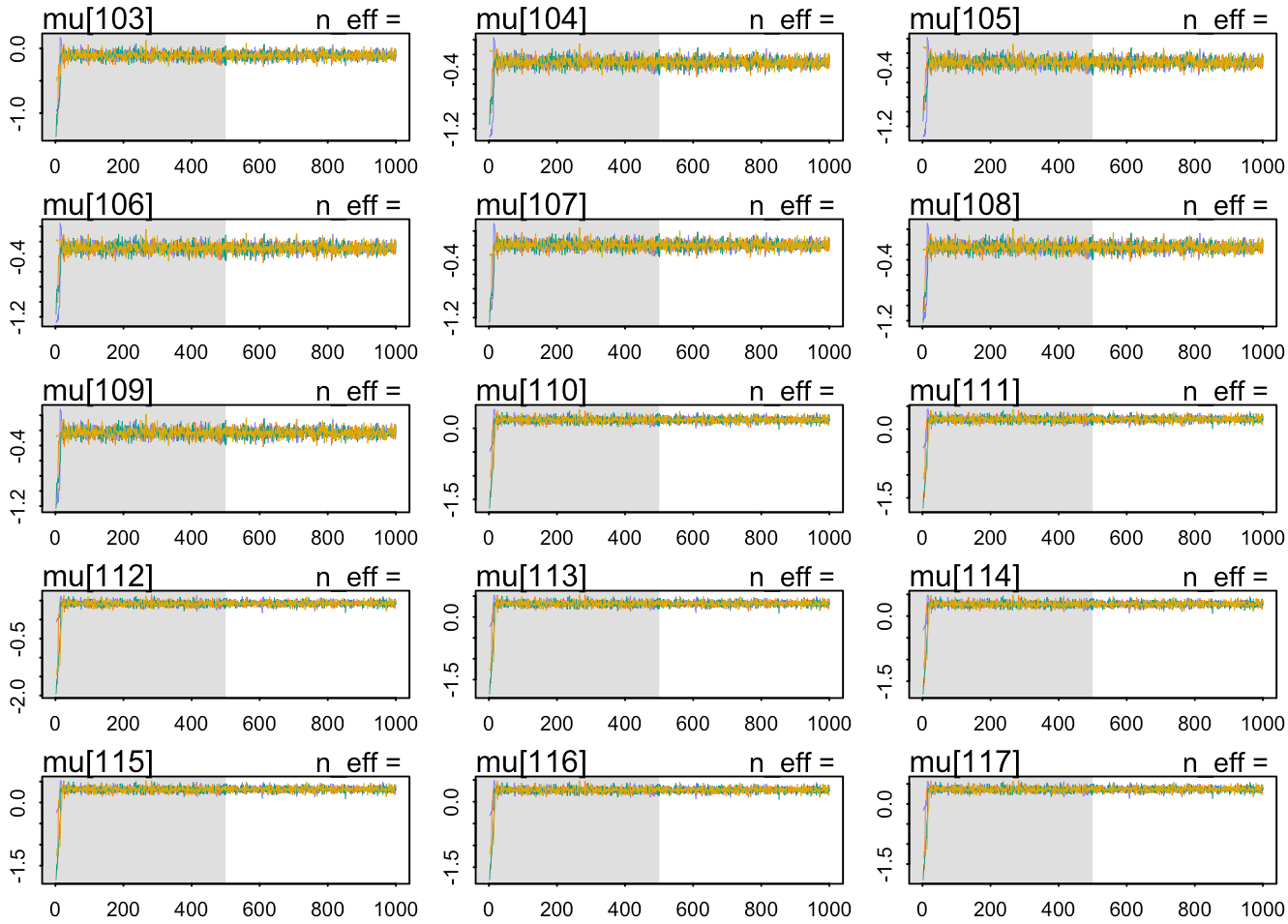
Waiting to draw page 7 of 11



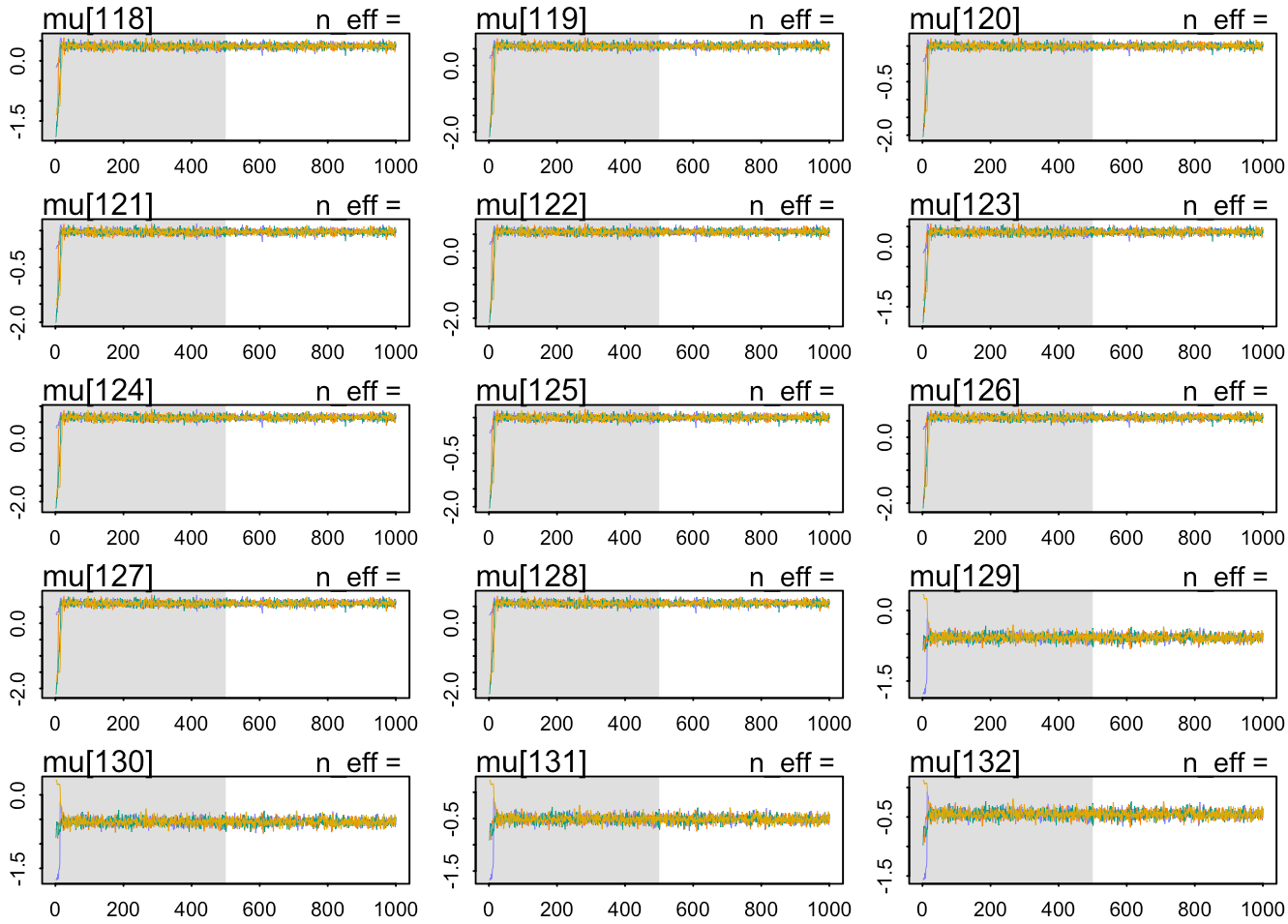
Waiting to draw page 8 of 11



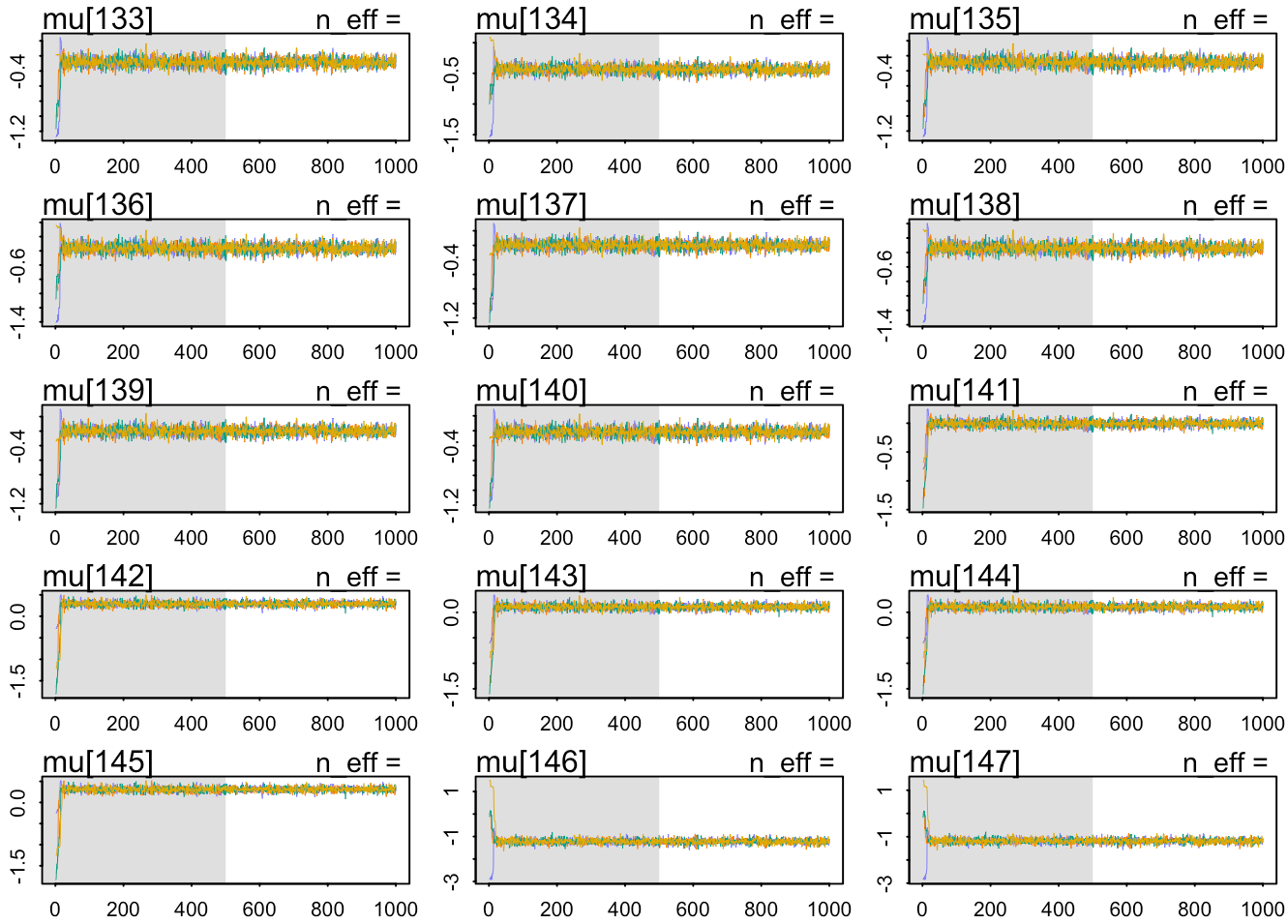
Waiting to draw page 9 of 11



Waiting to draw page 10 of 11

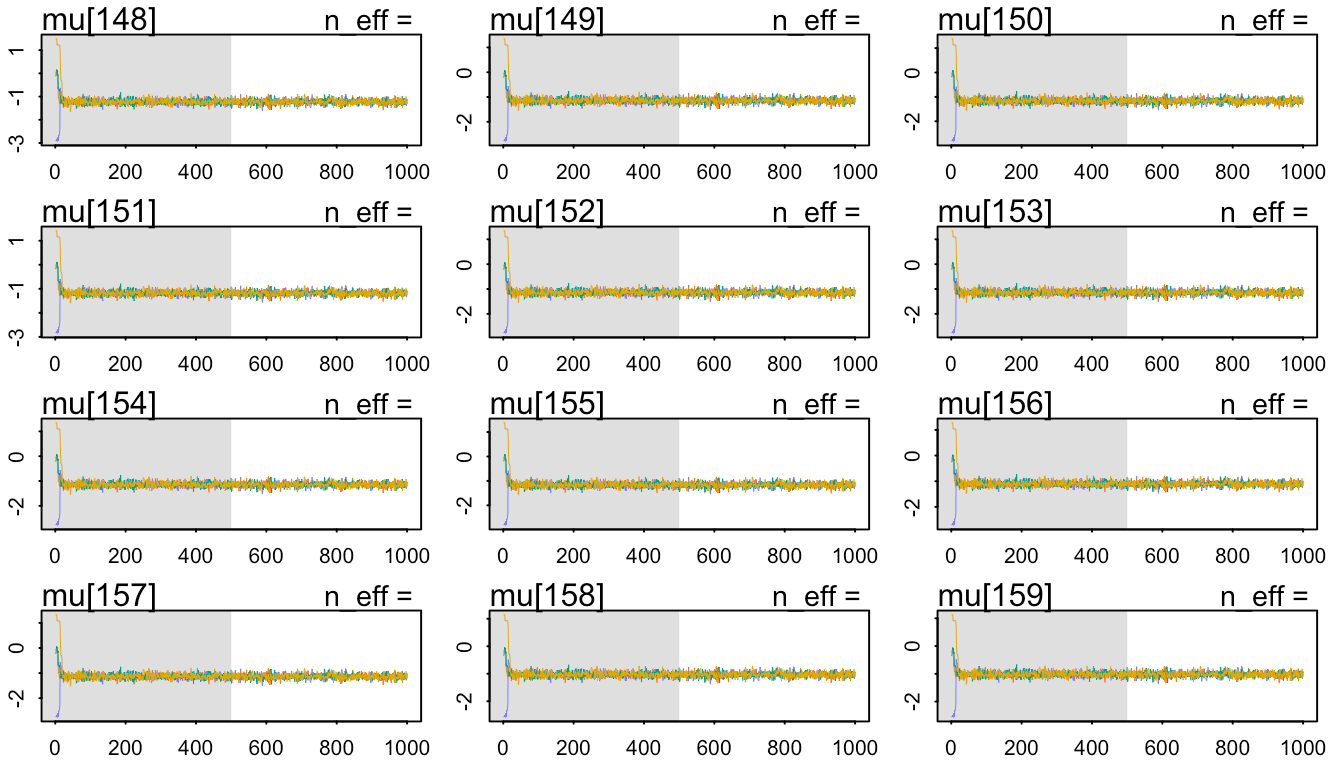


Waiting to draw page 11 of 11




```
WAIC(m2_mcmc)
```

```
      WAIC      lppd  penalty  std_err
1 341.2441 -164.8002 5.821849 40.44943
```



MCMC - Model 3: Length2 + Height -> Weight

```
# MCMC - Model 3: Length 2 + Height -> Weight
m3_mcmc <- ulam(
  alist(
    W~ dnorm( mu, sigma),
    mu <- a + bL * L + bH * H,
      a ~ dnorm(0, 1),
      bL ~ dnorm(0.5, 2),
      bH ~ dnorm(0.5, 2),
    sigma ~ dexp( 1)
), data = dcc,
   chains=4,
   cores=4,
   log_lik = TRUE
)
```

```
Removing one or more character or factor variables:

Species

Running MCMC with 4 parallel chains, with 1 thread(s) per chain...

Chain 1 Iteration:   1 / 1000 [  0%]  (Warmup)
```

```
Chain 1 Iteration: 100 / 1000 [ 10%]  (Warmup)
Chain 1 Iteration: 200 / 1000 [ 20%]  (Warmup)
Chain 1 Iteration: 300 / 1000 [ 30%]  (Warmup)
Chain 1 Iteration: 400 / 1000 [ 40%]  (Warmup)
Chain 1 Iteration: 500 / 1000 [ 50%]  (Warmup)
Chain 1 Iteration: 501 / 1000 [ 50%]  (Sampling)
Chain 1 Iteration: 600 / 1000 [ 60%]  (Sampling)
Chain 1 Iteration: 700 / 1000 [ 70%]  (Sampling)
Chain 1 Iteration: 800 / 1000 [ 80%]  (Sampling)
Chain 1 Iteration: 900 / 1000 [ 90%]  (Sampling)
Chain 1 Iteration: 1000 / 1000 [100%]  (Sampling)
Chain 1 Informational Message: The current Metropolis proposal is about to be rejected
because of the following issue:

Chain 1 Exception: normal_lpdf: Scale parameter is 0, but must be positive! (in
'/var/folders/ym/vpwglz617b78sp1lhzq6ydx00000gn/T/RtmpJMlyzU/model-3e9d6e3b742e.stan',
line 27, column 4 to column 29)

Chain 1 If this warning occurs sporadically, such as for highly constrained variable
types like covariance matrices, then the sampler is fine,

Chain 1 but if this warning occurs often then your model may be either severely ill-
conditioned or misspecified.

Chain 1

Chain 2 Iteration:   1 / 1000 [  0%]  (Warmup)
Chain 2 Iteration: 100 / 1000 [ 10%]  (Warmup)
Chain 2 Iteration: 200 / 1000 [ 20%]  (Warmup)
Chain 2 Iteration: 300 / 1000 [ 30%]  (Warmup)
Chain 2 Iteration: 400 / 1000 [ 40%]  (Warmup)
Chain 2 Iteration: 500 / 1000 [ 50%]  (Warmup)
Chain 2 Iteration: 501 / 1000 [ 50%]  (Sampling)
Chain 2 Iteration: 600 / 1000 [ 60%]  (Sampling)
Chain 2 Iteration: 700 / 1000 [ 70%]  (Sampling)
Chain 2 Iteration: 800 / 1000 [ 80%]  (Sampling)
Chain 2 Iteration: 900 / 1000 [ 90%]  (Sampling)
Chain 2 Iteration: 1000 / 1000 [100%]  (Sampling)
Chain 3 Iteration:   1 / 1000 [  0%]  (Warmup)
Chain 3 Iteration: 100 / 1000 [ 10%]  (Warmup)
Chain 3 Iteration: 200 / 1000 [ 20%]  (Warmup)
Chain 3 Iteration: 300 / 1000 [ 30%]  (Warmup)
Chain 3 Iteration: 400 / 1000 [ 40%]  (Warmup)
Chain 3 Iteration: 500 / 1000 [ 50%]  (Warmup)
Chain 3 Iteration: 501 / 1000 [ 50%]  (Sampling)
Chain 3 Iteration: 600 / 1000 [ 60%]  (Sampling)
Chain 3 Iteration: 700 / 1000 [ 70%]  (Sampling)
Chain 3 Iteration: 800 / 1000 [ 80%]  (Sampling)
Chain 3 Iteration: 900 / 1000 [ 90%]  (Sampling)
Chain 3 Iteration: 1000 / 1000 [100%]  (Sampling)
Chain 4 Iteration:   1 / 1000 [  0%]  (Warmup)
```

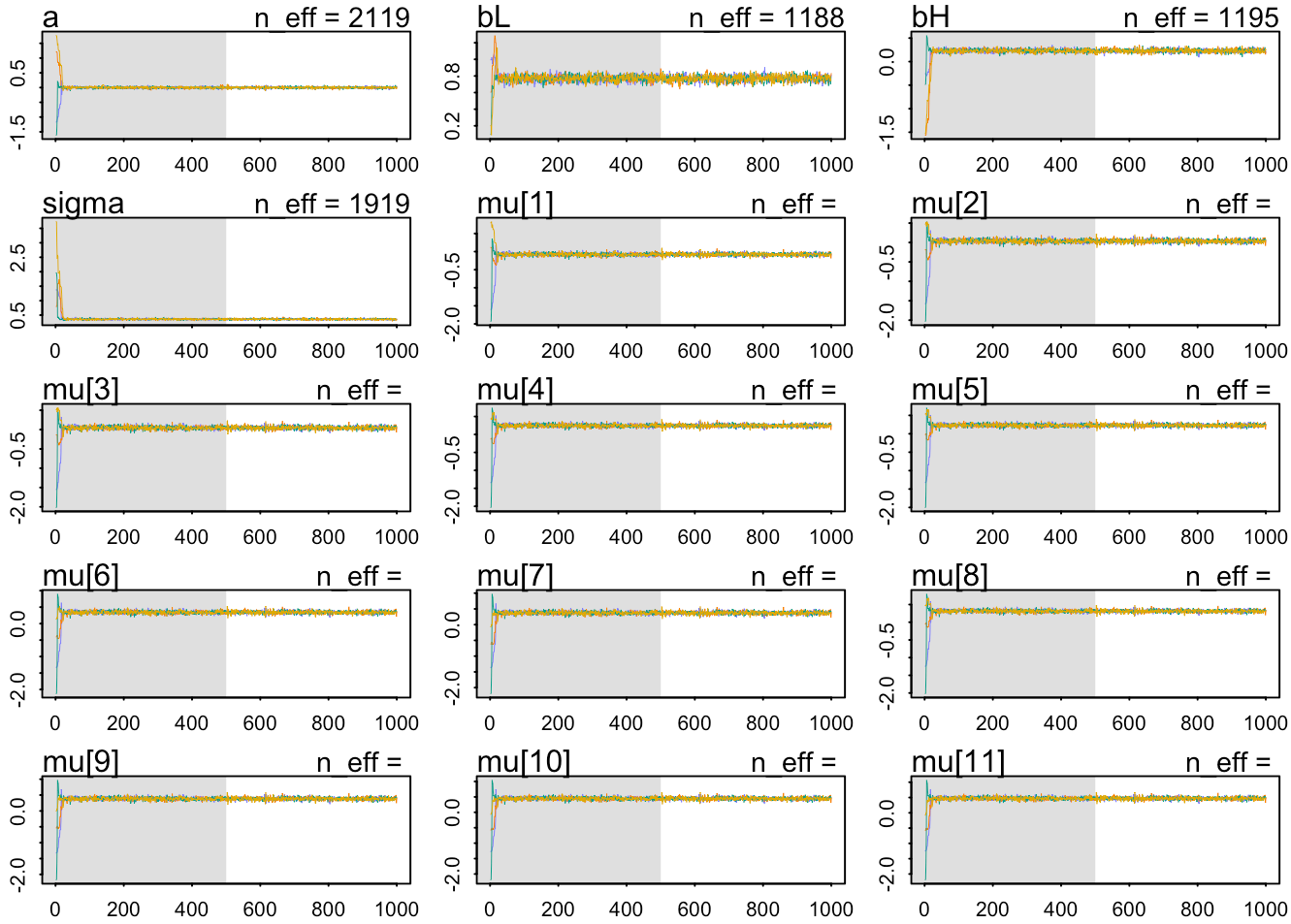
```
Chain 4 Iteration: 100 / 1000 [ 10%]  (Warmup)
Chain 4 Iteration: 200 / 1000 [ 20%]  (Warmup)
Chain 4 Iteration: 300 / 1000 [ 30%]  (Warmup)
Chain 4 Iteration: 400 / 1000 [ 40%]  (Warmup)
Chain 4 Iteration: 500 / 1000 [ 50%]  (Warmup)
Chain 4 Iteration: 501 / 1000 [ 50%]  (Sampling)
Chain 4 Iteration: 600 / 1000 [ 60%]  (Sampling)
Chain 4 Iteration: 700 / 1000 [ 70%]  (Sampling)
Chain 4 Iteration: 800 / 1000 [ 80%]  (Sampling)
Chain 4 Iteration: 900 / 1000 [ 90%]  (Sampling)
Chain 4 Iteration: 1000 / 1000 [100%]  (Sampling)
Chain 4 Informational Message: The current Metropolis proposal is about to be rejected
because of the following issue:

Chain 4 Exception: normal_lpdf: Scale parameter is 0, but must be positive! (in
'/var/folders/ym/vpwglz617b78sp1lhzq6ydx00000gn/T/RtmpJMlyzU/model-3e9d6e3b742e.stan',
line 27, column 4 to column 29)

Chain 4 If this warning occurs sporadically, such as for highly constrained variable
types like covariance matrices, then the sampler is fine,

Chain 4 but if this warning occurs often then your model may be either severely ill-
conditioned or misspecified.

Chain 4

Chain 1 finished in 0.1 seconds.
Chain 2 finished in 0.1 seconds.
Chain 3 finished in 0.1 seconds.
Chain 4 finished in 0.1 seconds.

All 4 chains finished successfully.
Mean chain execution time: 0.1 seconds.
Total execution time: 0.2 seconds.
```

```
precis(m3_mcmc)
```

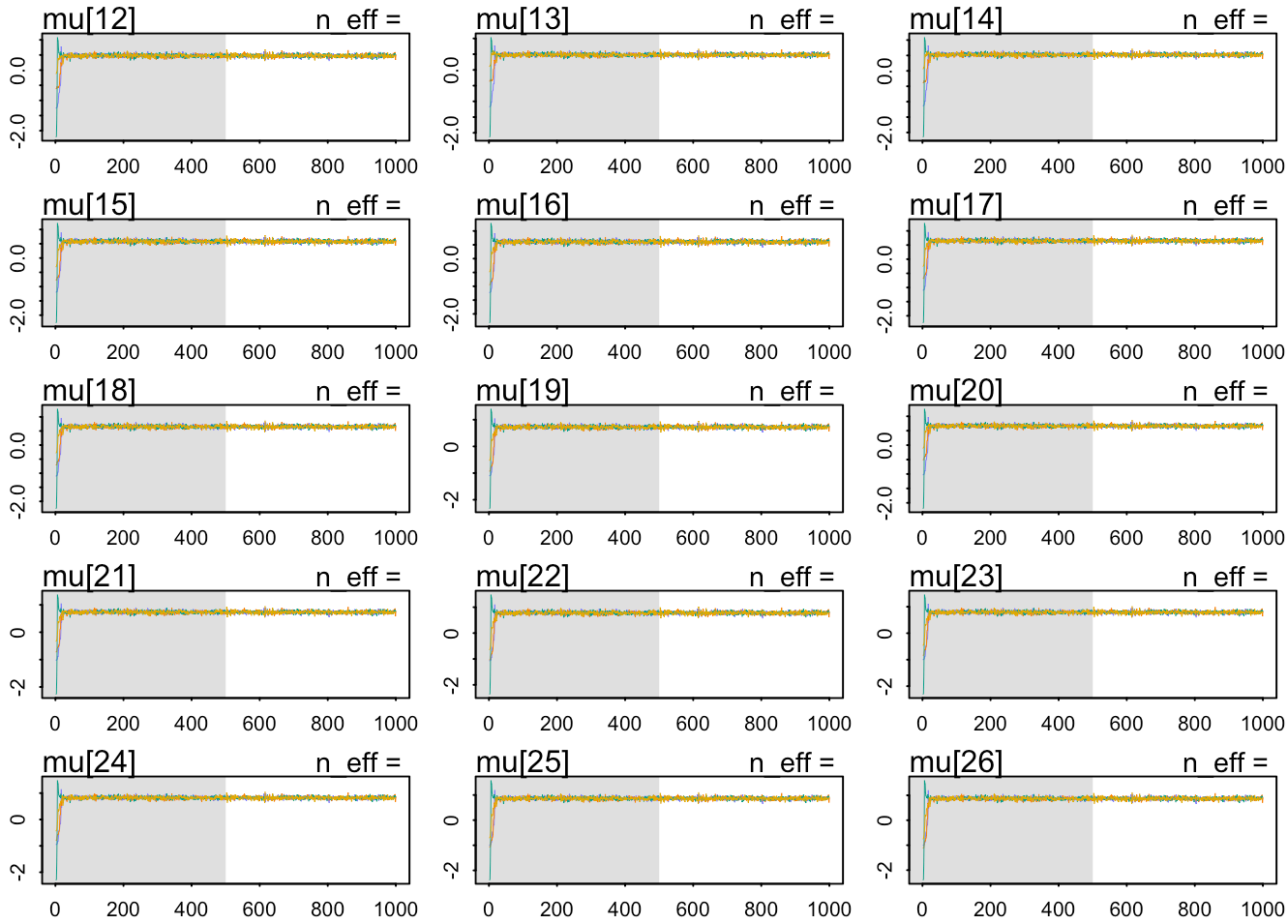
```
              mean         sd        5.5%      94.5%     rhat ess_bulk
a     0.0006657714 0.02821998 -0.04261186 0.04411038 1.000594 2119.008
bL    0.7698415090 0.03654280  0.70979565 0.82783548 1.002754 1188.206
bH    0.2317047609 0.03697904  0.17493946 0.29127449 1.001699 1194.819
sigma 0.3577940340 0.02198468  0.32540090 0.39522983 1.000592 1918.663
```

```
traceplot(m3_mcmc)
```

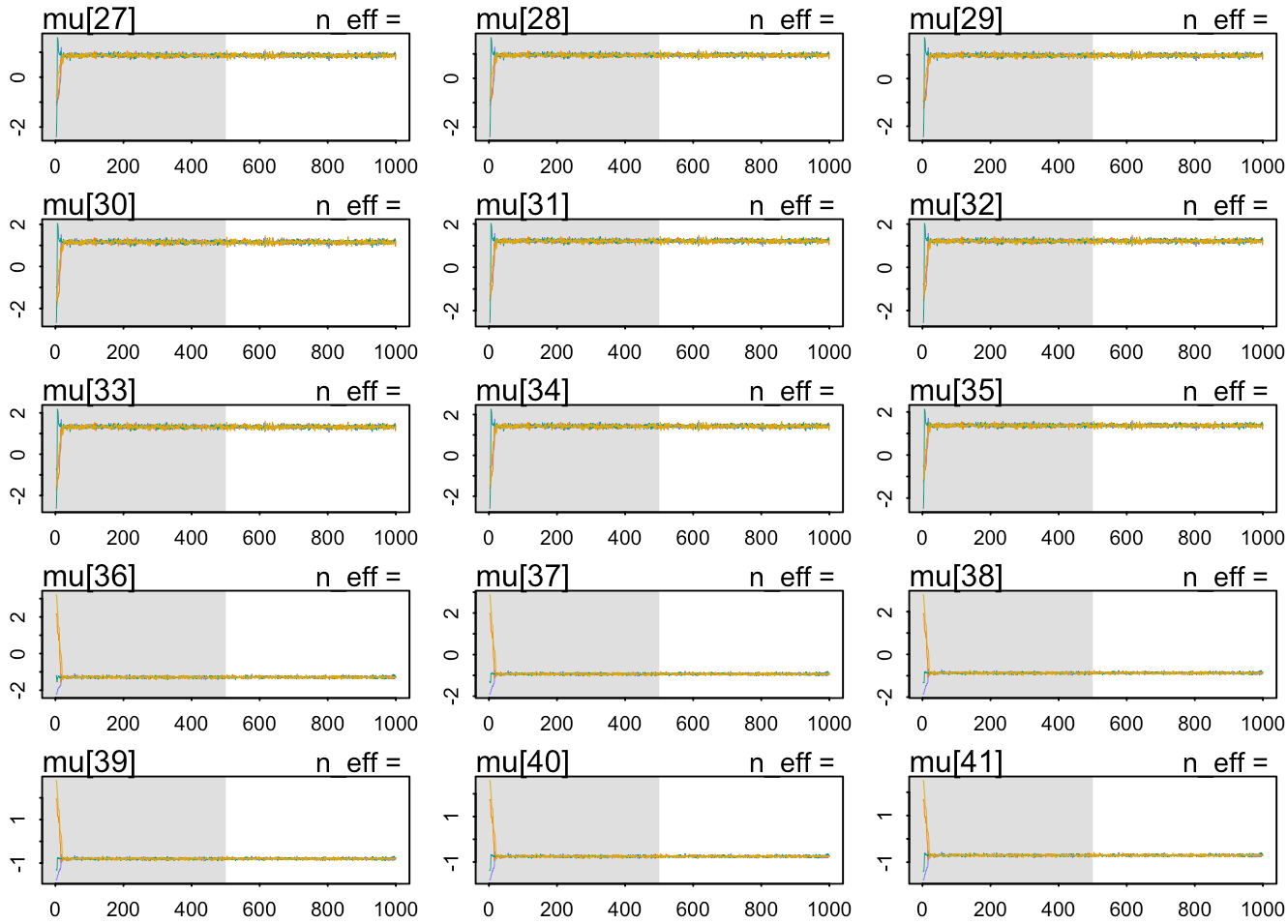
```
Waiting to draw page 2 of 11
```



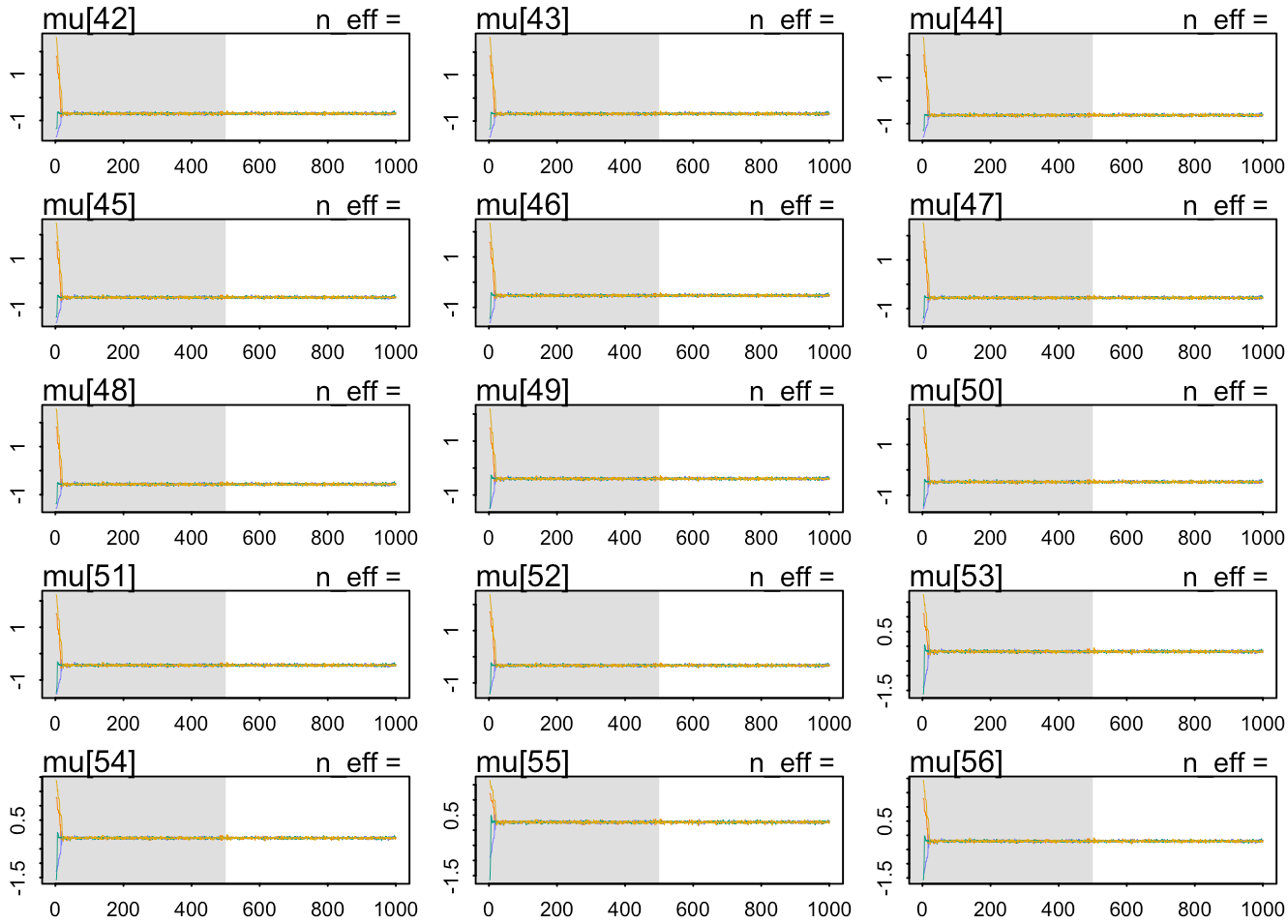
Waiting to draw page 3 of 11



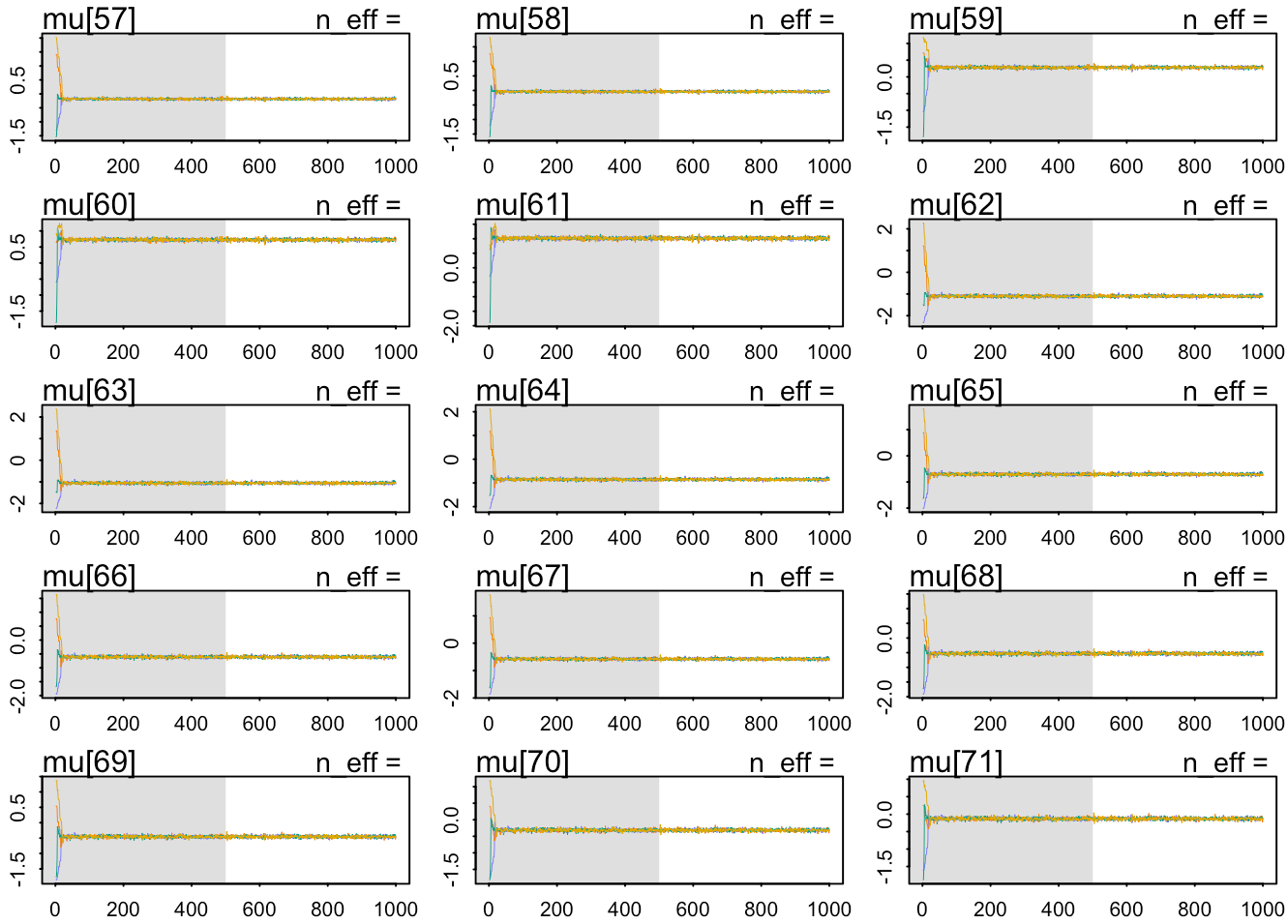
Waiting to draw page 4 of 11



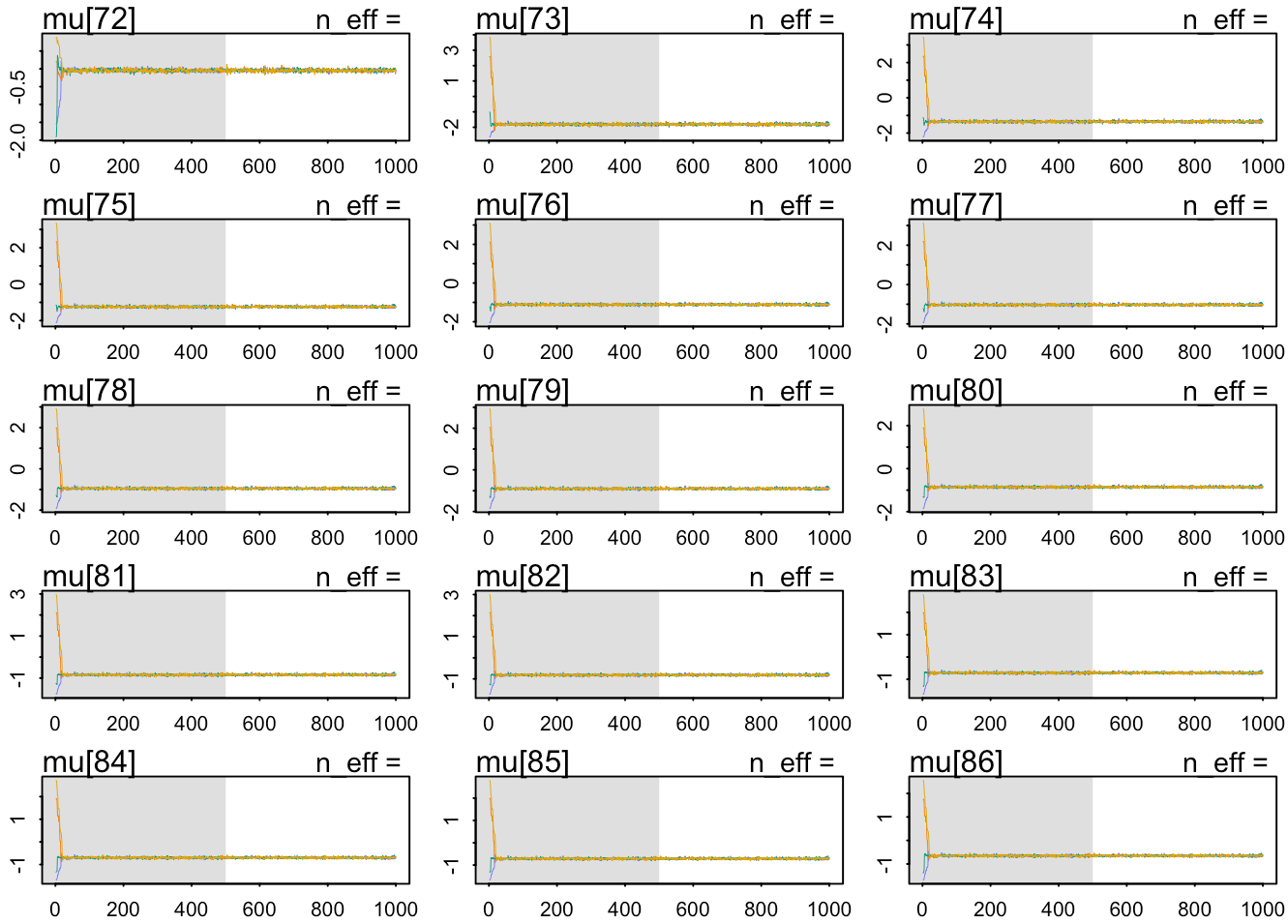

```
Waiting to draw page 5 of 11
```



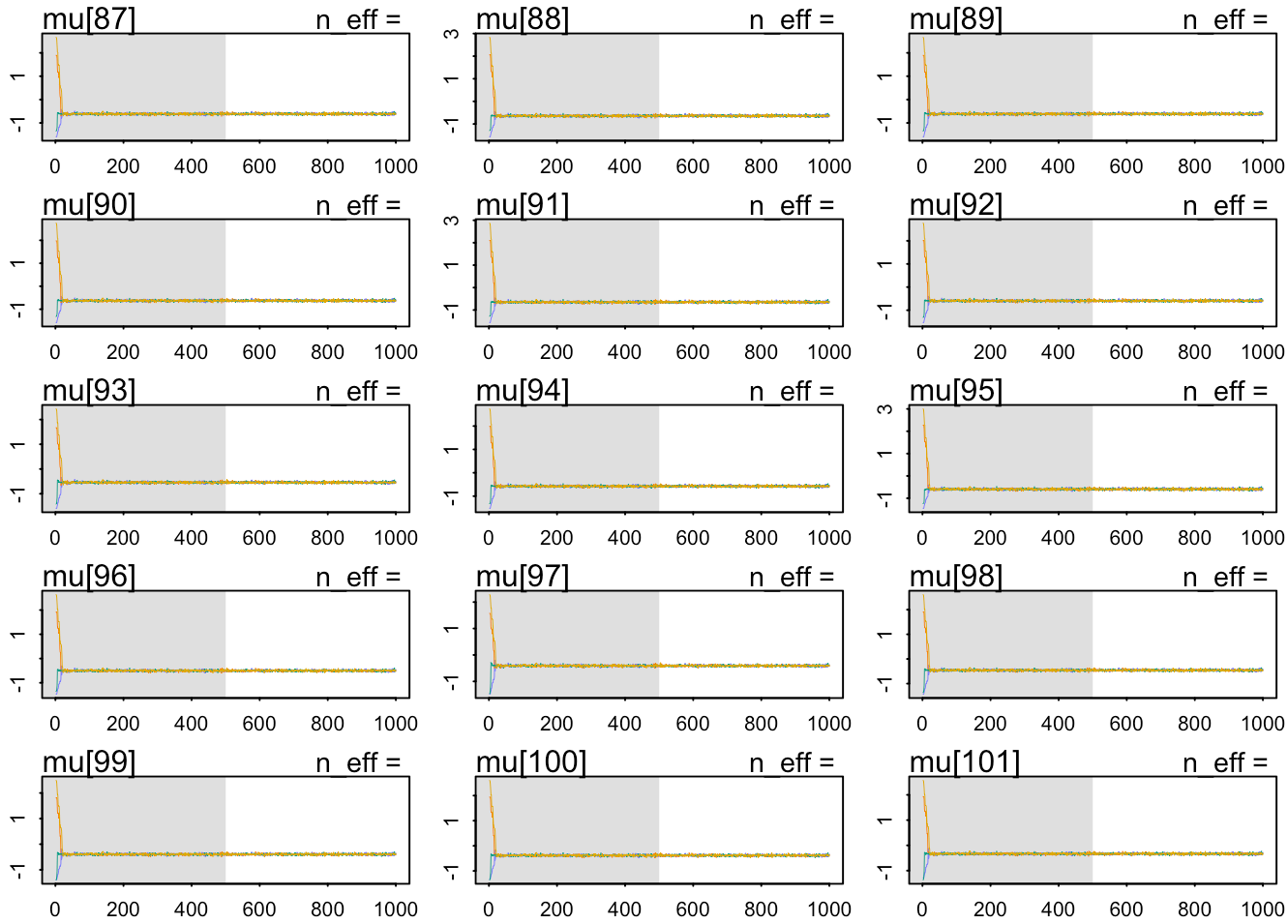
Waiting to draw page 6 of 11



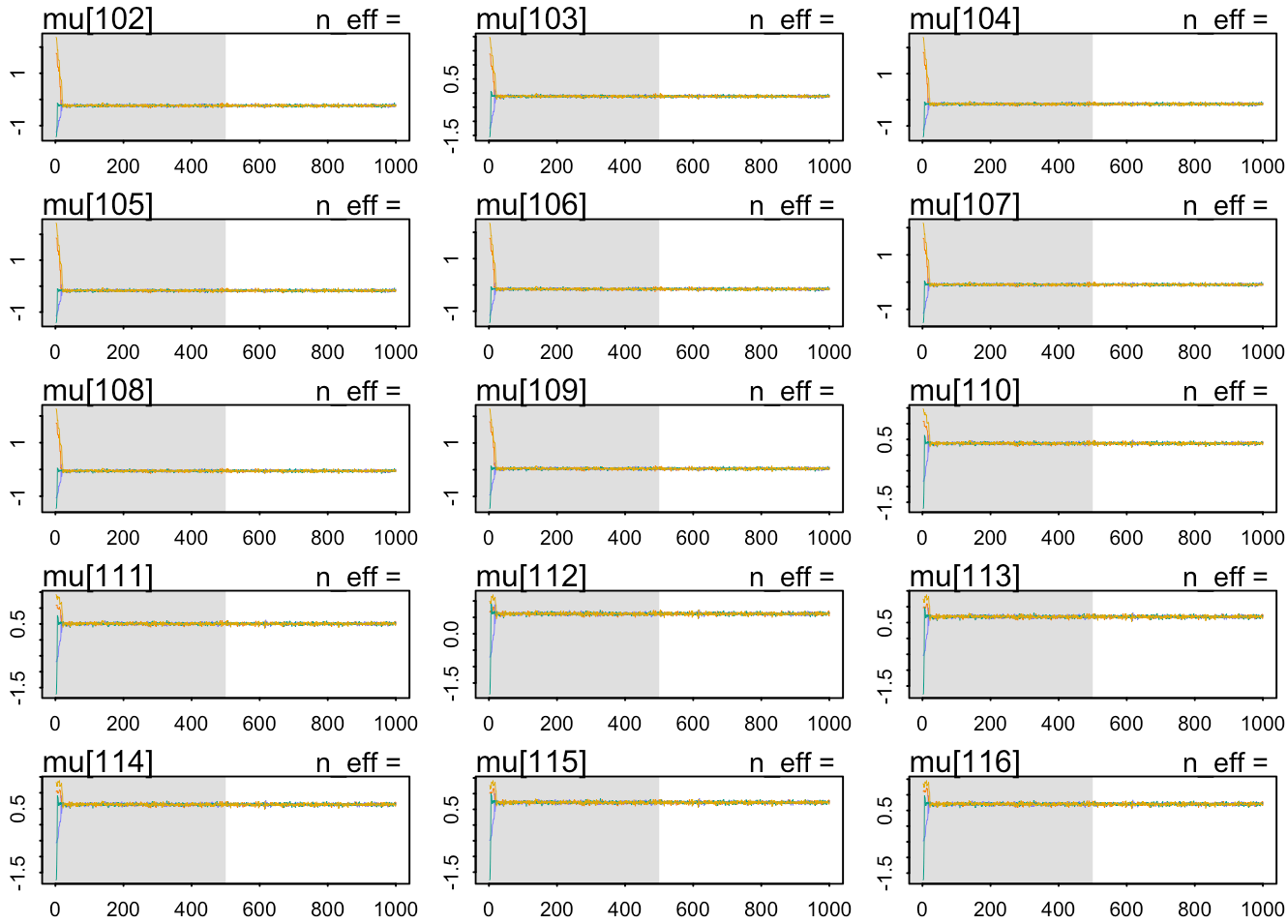
Waiting to draw page 7 of 11



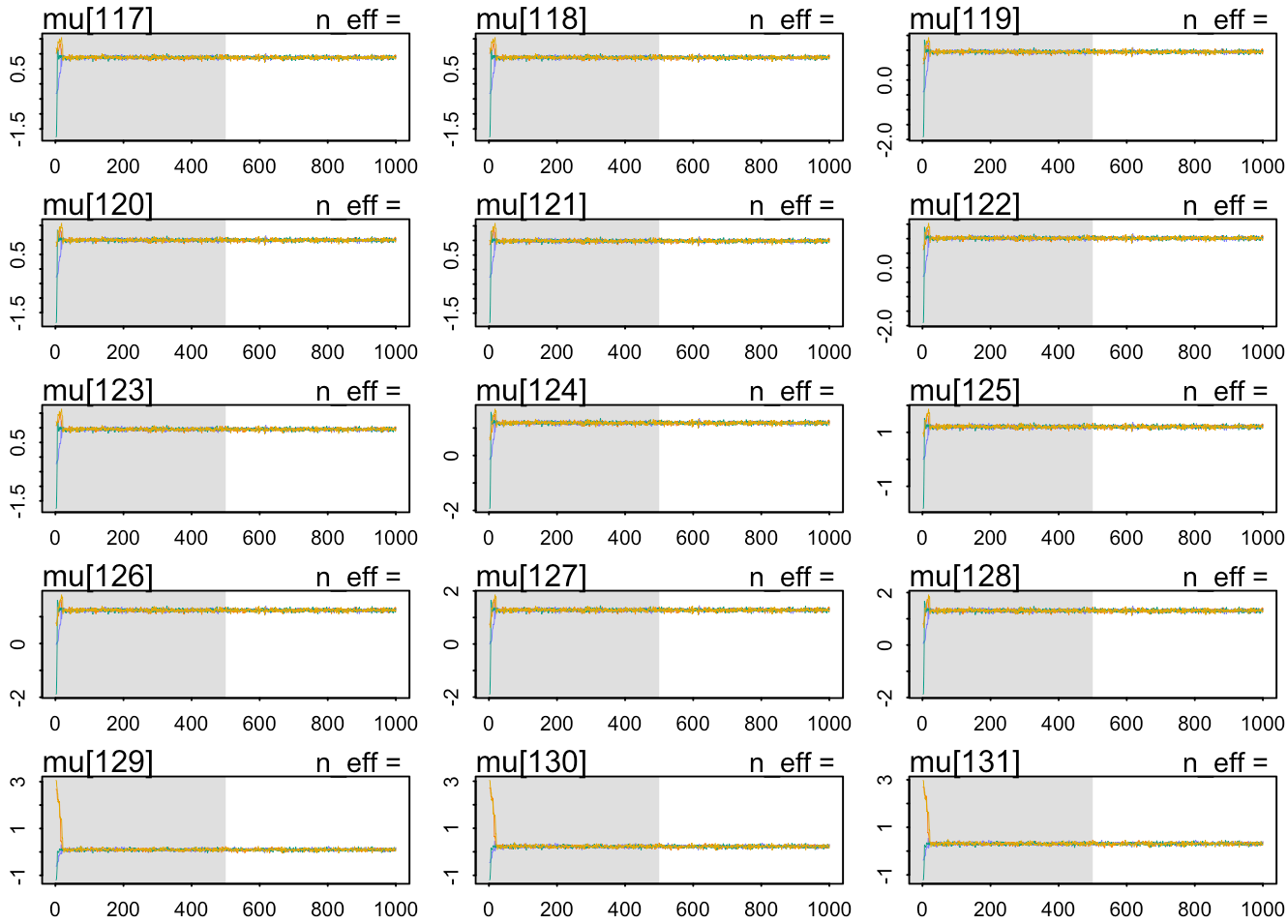
Waiting to draw page 8 of 11



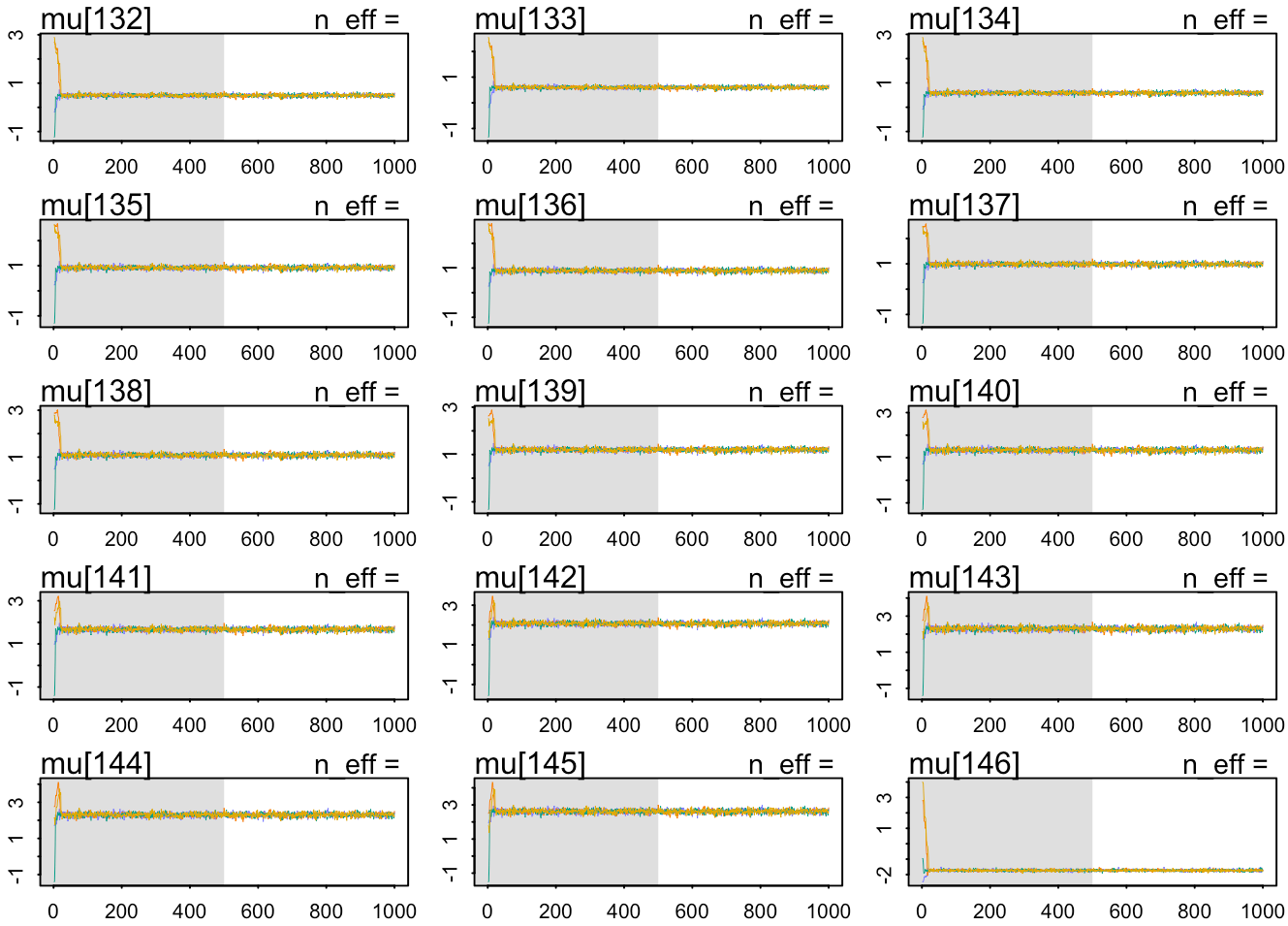
Waiting to draw page 9 of 11



Waiting to draw page 10 of 11

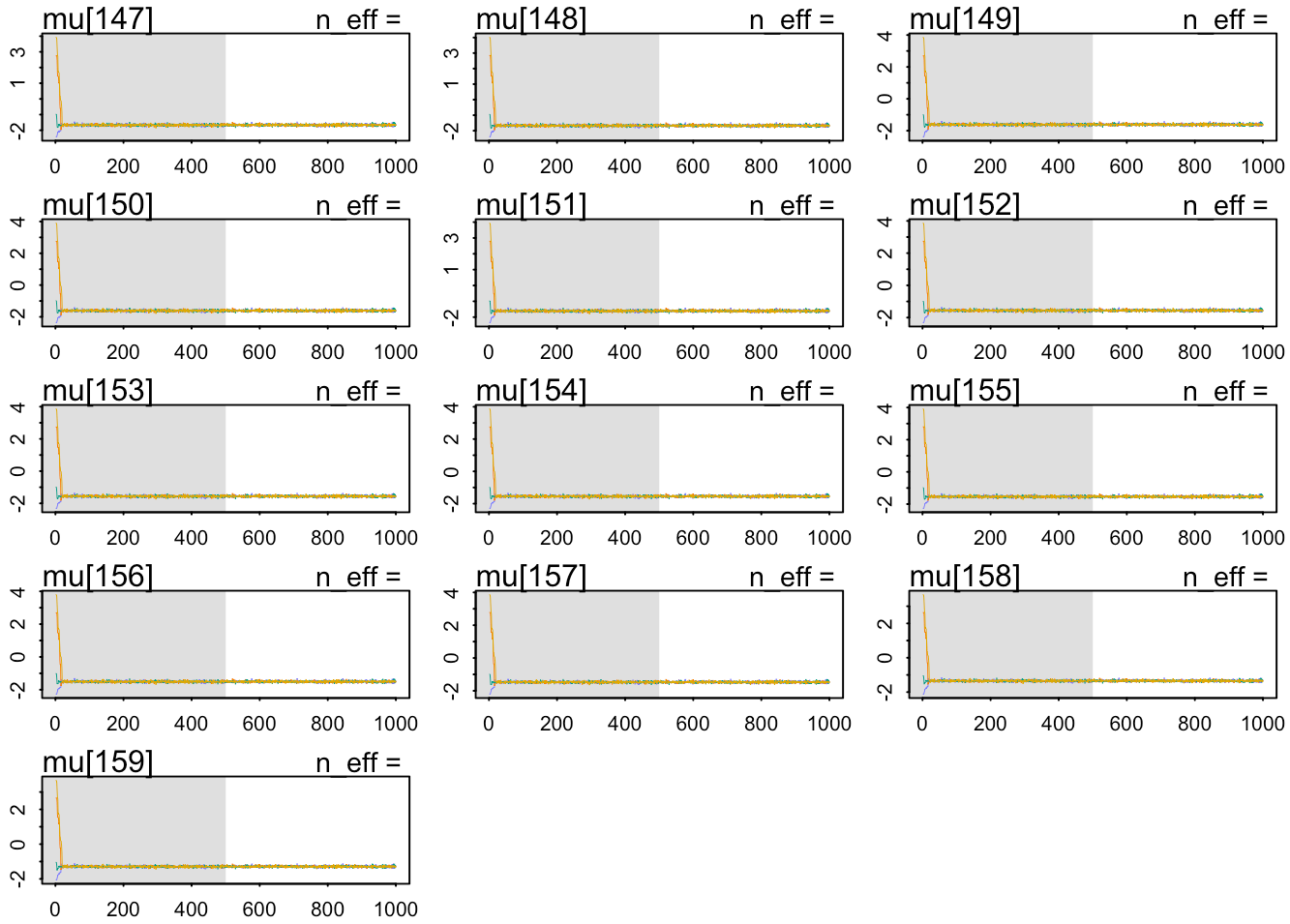


Waiting to draw page 11 of 11




```
WAIC(m3_mcmc)
```

```
      WAIC      lppd  penalty std_err
1 130.2759 -58.77563 6.362314 20.1096
```



Comparing the Models using WAIC

```
compare( m1_mcmc , m2_mcmc , m3_mcmc )
```

```
           WAIC       SE     dWAIC      dSE    pWAIC       weight
m3_mcmc 130.2759 20.10960   0.00000       NA 6.362314 9.999999e-01
m1_mcmc 162.2902 20.43395  32.01436 15.39275 3.991606 1.117300e-07
m2_mcmc 341.2441 40.44943 210.96826 31.54540 5.821849 1.544631e-46
```

The **WAIC (Widely Applicable Information Criterion)** values indicate the relative performance of the three models, with lower values representing better predictive accuracy and model fit.

Explanation:

- **M1 (WAIC = 162.0)**:
  Using length alone provides a relatively good fit, with a lower WAIC than M2. Length is a strong predictor of weight.

- **M2 (WAIC = 341.3)**:
  Using height alone results in the worst fit. The high WAIC indicates that height is less effective at predicting weight compared to length or the combined model.

- **M3 (WAIC = 130.4)**:
  Combining length and height provides the best fit. The lowest WAIC suggests that including both predictors captures more variability in weight and improves predictive performance compared to using either variable alone.

Length is a stronger individual predictor of weight than height (as seen in M1 vs. M2), but combining both predictors in M3 gives the best overall model fit.

Resources Used/ Works Cited

1. Fish Market Data (https://www.kaggle.com/datasets/vipullrathod/fish-market/data)
2. DAG Models (https://cran.r-project.org/web/packages/ggdag/vignettes/intro-to-dags.html)